ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

10

Practique Matemáticas y Estadística con el ordenador

Jesús Salcedo

SINCLAIR AMSTRAD COMMODORE IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

10

Practique Matemáticas y Estadística con el ordenador

EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S.A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Directores de la colección:
MANUEL ALFONSECA, Doctor Ingeniero de Telecomunicación y Licenciado en Informática
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño y maquetación:
BRAVO-LOFISH.

Dibujos:
JOSE OCHOA Y ANTONIO PERERA.

Tomo X. **Practique Matemáticas y Estadística con el ordenador.**
JESUS SALCEDO, Asesor Informático

Ediciones Siglo Cultural, S.A.

Dirección, redacción y administración:
Sor Angela de la Cruz, 24-7.º G. Teléf. 279 40 36. 28020 Madrid.

Publicidad:
Gofar Publicidad, S.A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:
COEDIS, S.A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América. 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-034-0.
ISBN de la obra: 84-7688-018-9.

Fotocomposición:
ARTECOMP, S.A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. Pinto (Madrid).

Depósito legal: M-39893-1986.
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S.A.
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid

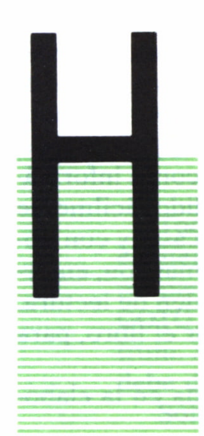
Octubre, 1986.
P.V.P. Canarias: 365,-

INDICE

Los programas que aparecen en este libro funcionan en los ordenadores:

IBM–PC, XT, AT y compatibles.
AMSTRAD–464, 664, 6128, 1512.
SINCLAIR–SPECTRUM 48 K, 128 K, PLUS, PLUS 2.
MSX–Todos los modelos.
COMMODORE–CBM 64 y CBM 128.

INTRODUCCION



HASTA la aparición del ordenador, el hombre malgastó mucho tiempo en realizar operaciones de cálculo y matemáticas de todo tipo, así como grandes procesos rutinarios. El llamado «cerebro electrónico» agilizó todas estas acciones. Sin embargo, las limitaciones de los ordenadores son grandes.

Queremos dejar constancia de la «torpeza» del ordenador. A lo largo del libro se subrayará este «defecto» de los microordenadores. Una computadora no resuelve un problema, sino que ayuda a su resolución. Hay que aprovechar su mayor cualidad: la gran velocidad operacional.

Con la masificación del uso de los ordenadores, más concretamente de los personales, el mito de máquina terrible e inabordable parece haber desaparecido. Un micro, mini o gran ordenador no hace más de lo que le mandemos; su capacidad de decisión propia es nula, e incluso es de desear (es una opinión personal) que no llegue nunca a tenerla.

Lógicamente la aplicación más directa de un microordenador son las Matemáticas y la Estadística. Desde que Pascal y Leibniz, con sus ingenios mecánicos precursores de las actuales calculadoras, se dieran cuenta de la necesidad de descargar este pesado tabajo operacional, en una máquina para profundizar más en la Ciencia, continuando por Babbage con sus famosas máquinas, diferencial y analítica, hasta hoy día, en que el ordenador ha invadido nuestras vidas, está clara la utilidad de una herramienta capaz de realizar estas monótonas e interminables operaciones, *en aplicaciones matemáticas*.

A lo largo de todo este libro abordaremos diferentes temas, en los que nuestro ordenador personal puede echarnos una mano. Habrá, naturalmente, capítulos en los que la labor del ordenador sea más agradecida, en otros su mérito se deberá más al programador.

En todo caso se ha tratado de dar un repaso general de materias matemáticas y estadísticas, algunas de ellas poco usuales, procurando valernos de nuestro ordenador casero; esperamos que con ello vaya adquiriendo el lector el hábito de utilizar su ordenador personal para abordar estas materias a partir de ahora.

Hemos introducido indicaciones, al lector más interesado, de cómo poder ampliar las aplicaciones aquí dadas. Además las matemáticas, queramos o no queramos, es, quizá, la ciencia más utilizada a diario, pudiendo convertir en algo más que una curiosidad el conocimiento de los temas tratados.

Ha sido ardua la tarea de seleccionar, por razones de espacio, sólo algunos temas entre la amplia gama de posibilidades matemáticas y estadísticas. La elección ha estado condicionada a presentar un poco de todo, abarcando desde los conjuntos hasta las integrales definidas, procurando no profundizar demasiado en cada tema tanto matemática como informáticamente. Este libro está ideado para que pueda ser entendido fácilmente por un lector con una base media matemática (e incluso escasa) y con nociones elementales del microordenador y del BASIC. Los temas son, en general, independientes uno de otro, pudiéndose alterar el orden de lectura. De todas formas aconsejaríamos seguir la línea marcada por el libro, para su mejor comprensión.

Finalmente agradecer a amigos y colaboradores su ayuda en la elaboración de este libro.

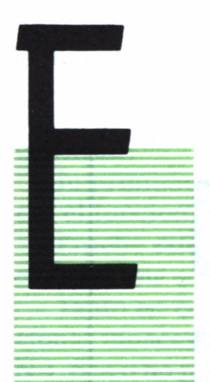


NOTA SOBRE LOS PROGRAMAS

Todos los programas del libro están realizados con un BASIC elemental para que sean válidos para la mayoría de los ordenadores personales. Se emplean siempre instrucciones elementales (IF, GOTO, etc.). Están probados en un Amstrad 6128 y en un Spectrum. Además se han revisado manualmente para Commodore. No debe dar problemas en cualquier otro micro.

Sólo se emplean instrucciones gráficas (PLOT, DRAW, etc.) en tres programas: estas instrucciones son las que más varían de un microordenador a otro, y quizá le exijan hacer alguna pequeña sustitución en su microordenador. De todas formas, se explica el planteamiento de los diferentes gráficos, facilitando así la labor de conversión.

DIVISIBILIDAD 1



L uso de las calculadoras ha originado un desuso de las operaciones aritméticas básicas realizadas manualmente.

En este capítulo trataremos una de ellas: la división. No vamos a enseñar al ordenador a dividir, esto ya lo sabe, sino que intentaremos auxiliarnos con nuestro equipo para ciertos cálculos reiterativos en los que intervenga la división. Nos referimos a la descomposición en factores primos, M.C.D., M.C.M. y la aplicación práctica de éstos. Los conceptos introducidos son muy sencillos: seguro que usted los conoce. Sin embargo, si no en este momento quizá sí en otro de este libro, le resultará divertido repasar estas cifras teóricas para aprovechar al máximo todas las experiencias que le proponemos.

Empecemos por lo más simple, *saber si un número es primo.*

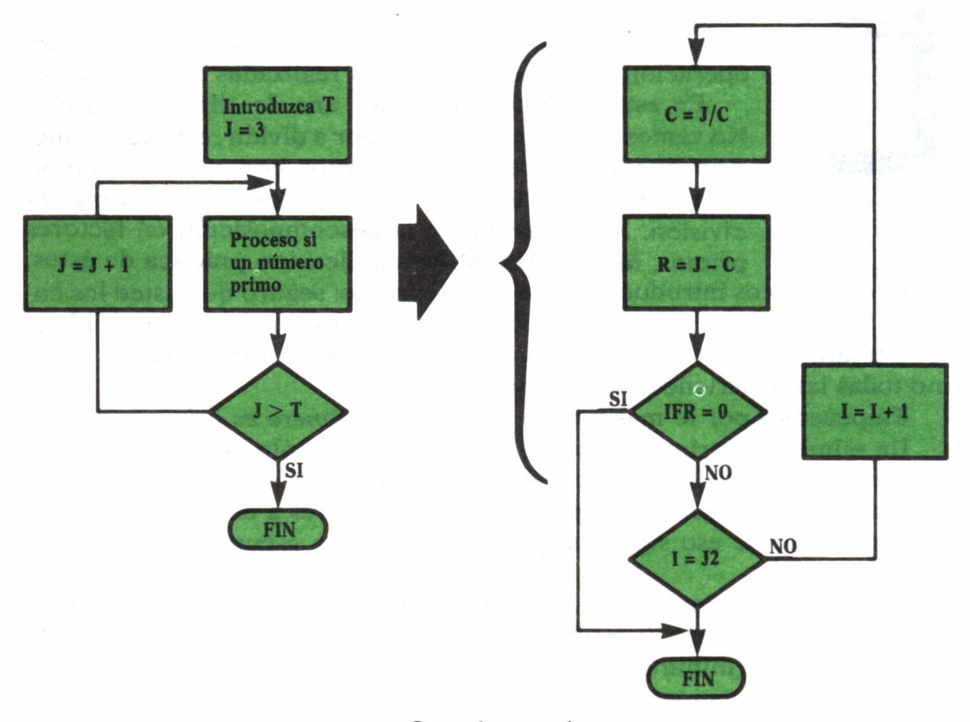
Un número es primo, según las Matemáticas, si sólo es divisible por él mismo y por la unidad. Sigamos en nuestro programa este planteamiento y dividamos el número dado por todos los inferiores a él. Si alguno lo divide exactamente eso significa que el número que estamos probando no es primo.

Eso es exactamente lo que hacemos en el programa que les ofrecemos a continuación: en las líneas 30 a 70 se va incrementando el índice J (línea 30) desde 2 (mínimo divisor que probamos hasta N-1 (el mayor) y se comprueba si el cociente N/J es entero o no (si N/J es entero, su parte entera –JNT(N/J)– coincidirá con él y al multiplicar por J en la línea 50 hará que R=0). Naturalmente, si N/J es entero, J es divisor de N y N no es número primo.

```
10 REM CALCULO SI UN NUMERO ES PRIMO
20 INPUT "NUMERO ELEGIDO: ";N
30 FOR J=2 TO N-1
```

```
40 LET C=INT(N/J)
50 LET R=N-J*C
60 IF R=0 AND N<>2 THEN PRINT "NO ES PRIMO":GOTO 9
99
70 NEXT J
80 PRINT "ES PRIMO"
999 END
```

Compliquemos un poco este sencillo programa. Calculemos ahora la *tabla de los números primos,* desde 1 hasta el número deseado. Observe el organigrama 1



Organigrama 1

y el programa:

```
10 REM CALCULO TABLA DE NUMEROS PRIMOS.POR FCO.
   MORALES
20 INPUT "HASTA QUE NUMERO? ";T
```

```
30 FOR I=1 TO T
40 FOR J=2 TO I
60 IF J>SQR(I) THEN GOTO 100
70 LET R=I-J*INT(I/J)
80 IF R=0 THEN GOTO 110
90 NEXT J
100 PRINT I;" ";
110 NEXT I
```

Como verá, lo único que hemos hecho es introducir nuestro programita dentro de un FOR, para que se repita el proceso tantas veces como indique el número introducido a modo de límite de la tabla que queremos construir.

Imprimimos el 1 y el 2, que son primos de antemano, para no complicar el funcionamiento de los bucles FOR.

Pruebe este programa y fíjese en que introduciendo un N muy grande el programa se hace muy lento. ¿Cómo ganar velocidad?

Tanto en este programa como en cualquier otro, hay que evitar operaciones innecesarias. Si miramos, se observa que no hace falta dividir hasta N-1. Sea o no primo, simplemente con repetir el proceso hasta la mitad bastaría. La razón es sencilla: a partir de N/2, es decir, desde la mitad en adelante, el cociente estaría comprendido entre 2 y 1, nunca poría ser exacto. Sólo podría ser exacto en N/2 y, por supuesto, en el extremo superior, N. Procedamos a la reforma:

```
36 FOR I=2 TO N/2
```

Vuelva a probarlo y comprobará la diferencia de tiempo.

Ya hemos conseguido saber si un número es primo, ahora sigamos el mismo razonamiento para *descomponer un número en sus factores primos*. Mire el programa:

```
10 REM DESCOMPOSICION EN FACTORES PRIMOS.POR FCO.
MORALES
20 INPUT "NUMERO ELEGIDO ";N
30 LET N1=N
40 LET I=2
50 IF I>SQR(N) THEN GOTO 150
60 IF INT (N/I)<>N/I THEN GOTO 100
```

```
70 PRINT I;" ";
80 LET N=N/I
90 GOTO 50
100 IF I=2 THEN GOTO 130
110 LET I=I+2
120 GOTO 50
130 LET I=3
140 GOTO 50
150 PRINT N;" ";
160 IF N1>N THEN 999
170 PRINT "ES UN NUMERO PRIMO"
180 GOTO 999
190 PRINT " SON LOS FACTORES DE ";N1
999 END
```

Las instrucciones 40, 50 y 60 ya nos son familiares. En cambio, la condición del IF ha cambiado. Ahora se interroga si el resto (R) es distinto de cero. ¿Para qué?

Si nosotros directamente incrementamos el índice (J), después de dividir, en el caso de factor repetido no funcionaría. Lo detectaría la primera vez, pero al incrementar, ya no volvería a utilizarlo. Con este error resultaría que el cociente final (C), que debería ser 1, daría erróneo, o los factores resultantes no serían primos.

Para ello con las instrucciones 66 y 67 volvemos a repetir el proceso para un factor primo encontrado, y no incrementará hasta haber agotado todas las posibilidades con él.

¿Cuándo terminará el proceso?

Matemáticamente sería cuando el último cociente fuese 1, pero esta condición no nos bastaría.

Al despreciar los decimales (INT), podría darse el caso en el que la división diese como cociente 1, sin tener en cuenta que, a lo mejor, no fuese exacta. Podríamos solucionarlo con:

```
IF R<> O THEN GOTO 70
```

Añadiendo la condición R=0, este caso sólo se daría en la última descomposición. Pero esto realmente se cumplirá también cuando N<J. Es decir, cuando el divisor sea mayor que el dividendo. A partir de este momento, las divisiones sucesivas serían inútiles, ya que el cociente (C) valdría siempre 0 y el resto 1. Con lo cual terminaría el programa cuando acabase el FOR.

Para el *cálculo del M.C.D. y el M.C.M. de dos números* nos basaremos en dos reglas matemáticas. Podríamos haber seguido el proceso clásico de descomponer los dos números en sus factores primos, para su posterior estudio.

Nos encontraríamos con una pega. Los factores habría que ir almacenándolos en algún sitio. La utilización de una tabla sería la solución más adecuada, pero estaría limitada a la dimensión (DIM) de la misma. Por tanto, no nos podemos arriesgar a quedarnos cortos ni a desaprovechar espacio.

Utilizaremos dos métodos más rápidos y elaborados, estudiados y demostrados matemáticamente; debido a ello nos fiaremos de su veracidad.

Para el M.C.D. el sistema utilizado consiste en dividir el mayor por el menor, si el resto de esta división es 0, el menor es el M.C.D. de los dos. En caso contrario, entramos en un proceso reiterativo, en el cual repetimos la operación de la siguiente manera:

El resto de la división se convierte en el divisor de la próxima y el divisor en el dividendo. Concluyendo, cuando el resto sea nulo, en este momento el último divisor es el M.C.D. de los dos números.

Esto es justamente lo que reflejamos en el programa:

```
10 REM CALCULO DEL M.C.D DE DOS NUMEROS
20 INPUT "INTRODUZCA EL PRIMER NUMERO ";A
30 INPUT "INTRODUZCA EL SEGUNDO NUMERO ";B
40 LET L=B:LET M=A
50 IF A>B THEN LET M=B:LET L=A
60 LET C=INT(L/M)
70 LET R=L-C*M
80 IF R=0 THEN GOTO 120
90 LET L=M
100 LET M=R
110 GOTO 60
120 PRINT "EL M.C.D. DE ";A;" Y ";B;" ES ";M
999 END
```

En las primeras instrucciones comprobamos cuál de los dos números es el mayor. A continuación, comienza el proceso reiterativo, que concluye cuando R=0 (en la 80).

En este momento el divisor (M) es el M.C.D.

Conociendo el M.C.D. *conseguir el M.C.M.* es muy simple, ya que

$$\text{M.C.M.} = \frac{A*B}{\text{M.C.D.}}$$

Incluyamos dos nuevas instrucciones y sustituyamos la 120.

```
115  P=A*B
116  MCM=P/M
120  PRINT "EL M.C.M. ES:"; MCM
```

Ya hemos conseguido saber si un número es primo, descomponer un número y conseguir el M.C.M. y M.C.D. de dos números.

Apliquemos nuestro éxito a una cuestión práctica, que por su elaboración, no por su dificultad, se hacía pesada. Me estoy refiriendo a la suma de quebrados. (Lógicamente la diferencia de quebrados se haría con numeradores negativos, pero el proceso general es igual.)

Vamos a calcular la *suma de dos quebrados.*

Para ello se seguirá el siguiente proceso:

1.º Se pedirán los datos (30 y 40).

2.º Se calcula el M.C.M. de los denominadores (50 a la 150). (Esto no produce ninguna dificultad, ya que se ha estudiado a lo largo del capítulo.)

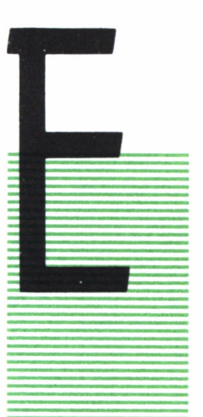
3.º Siguiendo las reglas matemáticas, obtenemos los numeradores (170 y 180). El resultado, lógicamente, será la suma de éstos (190).

```
10 REM SUMA DE DOS QUEBRADOS
20 DIM N(2):DIM D(2)
30 INPUT "INTRODUZCA LA PRIMERA FRACCION(N,D) ";N(
1),D(1)
40 INPUT "INTRODUZCA LA SEGUNDA FRACCION(N,D) ";N(
2),D(2)
50 LET L=D(2):LET M=D(1)
60 IF D(1)>D(2) THEN LET L=D(1):LET M=D(2)
70 LET C=INT(L/M)
80  LET R=L-C*M
90 IF R=0 THEN GOTO 120
100 LET L=M
110 LET M=R
120 LET P=D(1)*D(2)
130 LET MC=P/M
140 LET N(1)=N(1)*MC/D(1)
150 LET N(2)=N(2)*MC/D(2)
160 LET S=N(2)+N(1)
170 PRINT "LA SUMA ES ";S;"/";MC;:IF S/MC=INT(S/MC
) THEN PRINT "=";S/MC
999 END
```

Practique con varias fracciones y compruebe el resultado. ¡Espero que le vaya todo bien!

Como investigación para el lector podría intentar la suma de varias fracciones. Observe que no es más que ir sumándolas de dos en dos. Como «pista» le proponemos una posible solución: sería guardar los numeradores en una tabla y los denominadores en otra; sólo haría falta ir incrementando el índice e ir calculando la suma por pares. ¡Suerte!

POLINOMIOS 2



L estudio de los polinomios y sus posibles raíces se tratará a lo largo de este capítulo. En todos los programas se utiliza la misma filosofía en el tratamiento de los polinomios:

* Captura de datos: almacenando los coeficientes en una tabla de dimensiones N+1 (ya que el término independiente es necesario guardarlo).

* Tratamiento de un coeficiente: expandiéndolo a todos los demás gracias a las instrucciones FOR.

* Impresión del resultado: proceso similar al de la captura, sustituyendo el READ por PRINT.

Comencemos con lo más elemental: *INTRODUCIR LOS COEFICIENTES Y CALCULAR EL RESULTADO* para un valor dado.

```
10 INPUT "INTRODUZCA GRADO DEL POLINOMIO P: ";N
20 DIM P(N+1)
30 LET N=N+1
40 FOR J=1 TO N
50 PRINT "COEFICIENTE DE X^";J-1;":";
60 INPUT P(J)
70 NEXT J
80 INPUT "INTRODUZCA VALOR DE X DESEADO: ";X
90 FOR J=1 TO N
100 LET D=P(J)*(X^(J-1))
110 LET S=S+D
120 NEXT J
130 PRINT S
999 END
```

El programa es simple, y en él se aprecian, perfectamente, las tres fases explicadas al principio: Captura (10 a 60), tratamiento (65 a la 90) e impresión del resultado (100).

Superado este paso previo avancemos operando con los polinomios. Lo primero será la *suma*. La regla matemática es sumar los coeficientes del mismo grado (200).

```
10 REM SUMA DE POLINOMIOS. POR FCO. MORALES
20 INPUT "INTRODUZCA GRADO DEL POLINOMIO P1 :";N1
30 LET N1=N1+1
40 INPUT "INTRODUZCA GRADO DEL POLINOMIO P2 :";N2
50 LET N2=N2+1
60 LET N=N1
70 IF N2>N1 THEN LET N=N2
80 REM DIMENSIONAMOS MATRICES
90 DIM P(N):DIM Q(N):DIM S(N)
100 PRINT "POLINOMIO P1:"
110 FOR J=1 TO N1
120 PRINT "COEFICIENTE DE X^";J-1;
130 INPUT P(J)
140 NEXT J
150 PRINT "POLINOMIO P2:"
160 FOR J=1 TO N2
170 PRINT "COEFICIENTE DE X^";J-1;
180 INPUT Q(J)
190 NEXT J
200 PRINT:PRINT "LOS POLINOMIOS QUEDAN ASI:":PRINT
 "POLINOMIO P1:"
210 FOR J=1 TO N1
220 PRINT P(J);"X^";J-1;"+";
230 NEXT J
240 PRINT " 0 = 0":PRINT
250 PRINT "POLINOMIO P2:"
260 FOR J=1 TO N2
270 PRINT Q(J);"X^";J-1;"+";
280 NEXT J
290 PRINT " 0 = 0":PRINT
300 PRINT "LA SUMA DE LOS POLINOMIOS ES:"
310 FOR J=1 TO N
320 LET S(J)=P(J)+Q(J)
330 PRINT "EL COEFICIENTE DE X^";J-1;" ES ";S(J)
340 NEXT J
350 PRINT:PRINT "EL POLINOMIO S QUEDARA ASI:":PRINT
360 FOR J=1 TO N
370 PRINT S(J);"X^";J-1;"+";
380 NEXT J
390 PRINT "0 = 0"
999 END
```

Una vez conocida la suma pasaremos a la *multiplicación*. En ésta, el resultado se almacenará en una tabla de N+M. Cada elemento de una matriz multiplica a todos los de la otra, desplazando el resultado tantas posiciones como las que ocupa. Puede ocurrir que la posición donde vaya a dejar el resultado esté ocupada; en tal caso se suman los valores. Este proceso viene controlado por los FOR de las líneas 110 y 120. La operación y acumulación se da en la 130.

```
10 INPUT "INTRODUZCA GRADO DEL POLINOMIO P1: ";N1
15 LET N1=N1+1
20 INPUT "INTRODUZCA GRADO DEL POLINOMIO P2: ";N2
40 LET N2=N2+1
50 LET N=N1:IF N2>N1 THEN LET N=N2
70 REM DIMENSIONAMOS LAS MATRICES
80 DIM M(N1):DIM N(N2):DIM P(N1+N2)
90 PRINT "POLINOMIO P1:"
100 FOR J=1 TO N1
110 PRINT "COEFICIENTE DE X^";J-1
120 INPUT M(J)
130 NEXT J
140 PRINT "POLINOMIO P2:"
150 FOR J=1 TO N2
160 PRINT "COEFICIENTE DE X^";J-1
170 INPUT N(J)
180 NEXT J
190 FOR J=1 TO N1
200 FOR I=1 TO N2
210 LET P(I+J)=P(I+J)+(M(J)*N(I))
220 NEXT I
230 NEXT J
240 PRINT "LA MULTIPLICACION DE LOS POLINOMIOS
    ES:"
250 FOR J=2 TO N1+N2
260 PRINT "COEFICIENTE DE X^";J-2;" ES ";P(J)
270 NEXT J
999 END
```

La división general de polinomios es un proceso más complejo, pero puede abordarlo el lector a partir de un caso más sencillo: *LA REGLA DE RUFFINI*.

Esta se aplica en los casos de divisiones por un polinomio del tipo (X ± a). El polinomio cociente es de un grado menos; y sus coeficientes se calculan (100 a 120) a partir del anterior obtenido (S(J)). El primero es el coeficiente de grado N del polinomio (90) y a partir de él se obtendrán los demás.

Desarrollando el programa (que no presenta ninguna dificultad):

```
10 INPUT "INTRODUZCA NUMERO A DEL BINOMIO X-A: ";A
20 INPUT "INTRODUZCA GRADO DEL POLINOMIO: ";N
30 LET N=N+1
35 DIM P(N):DIM S(N-1)
40 FOR J=1 TO N
50 PRINT "COEFICIENTE X^";J-1
60 INPUT P(J)
70 LET S(J-1)=P(J)
80 NEXT J
90 FOR I=N-1 TO 2 STEP -1
100 LET S(I-1)=P(I)+A*S(I)
110 NEXT I
120 PRINT "LA SOLUCION ES: "
130 FOR J=N-1 TO 2 STEP -1
140 PRINT "COEFICIENTE DE X^";J-1;" ES ";S(J)
150 NEXT J
160 PRINT "EL RESTO ES: ";S(1)
999 END
```

Nos guiaremos de este programa para el cálculo de las *raíces enteras de un polinomio*. Lo primero que se hace es acotar éstas superiormente, es decir, obtener un número que sea superior a todas las raíces (cota superior). Según Laguerre, un número es *cota superior* de las raíces de un polinomio si al aplicar la regla de Ruffini todos los coeficientes del cociente y el resto son positivos. ¿Qué haremos?

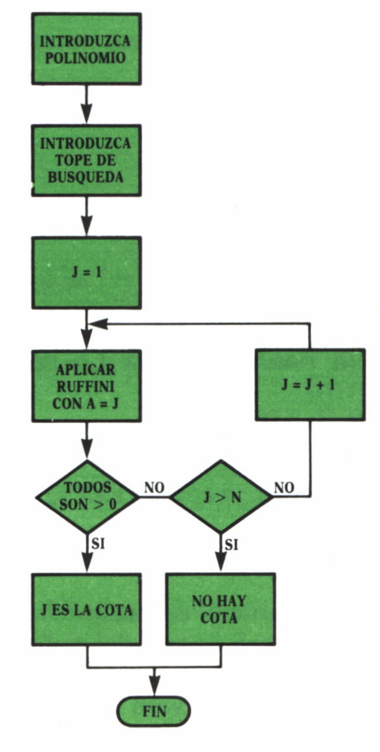
Para un valor de A, hay que aplicar la regla de Ruffini y analizar si el resultado es positivo (coeficientes y resto).

Si cumple la condición ya tenemos cota; en caso contrario, incrementaremos hasta conseguirlo.

```
10 INPUT "INTRODUZCA GRADO DEL POLINOMIO: ";N
20 LET N=N+1
30 DIM P(N):DIM S(N-1)
40 FOR J=1 TO N
50 PRINT "COEFICIENTE DE X^";J-1;
60 INPUT P(J)
70 NEXT J
80 INPUT "INTRODUZCA TOPE DE PRUEBAS: ";T
90 FOR K=1 TO T
100 LET S(N-1)=P(N)
110 FOR I=N-1 TO 2 STEP -1
```

```
120 LET S(I-1)=P(N)+K*S(I)
130 NEXT I
140 REM COMPROBACION SI TODOS LOS DATOS SON POSITI-
VOS
150 FOR J=N-1 TO 1 STEP -1
160 IF S(J)<0 THEN GOTO 190
170 NEXT J
180 PRINT "EL VALOR ";K;" ES COTA SUPERIOR":GOTO 9
99
190 NEXT T
200 PRINT "HASTA EL VALOR PROBADO NO HAY COTA"
999 END
```

Una vez lograda la cota, si es que existe, intentaremos *hallar las raíces*. El método que utilizan los estudiantes es ir probando número a número hasta conseguirlo. Si existe una raíz entera, ésta debe ser divisor del término independiente. Observe el organigrama 2



Organigrama 2

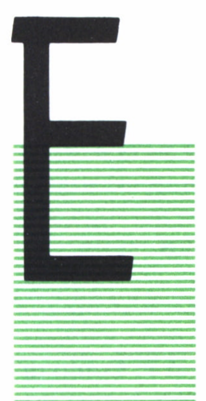
Si se fija tenemos que aplicar la descomposición de un número en sus factores primos (TEMA-2), para elegir después los que sean menores que la cota. Con éstos probamos uno a uno hasta conseguir hallar las raíces. El programa completo quedaría:

```
10 INPUT "INTRODUZCA GRADO DEL POLINOMIO: ";N
20 LET N=N+1:LET L=0:LET Z=0
30 DIM P(N):DIM S(N-1)
40 FOR J=1 TO N
50 PRINT "COEFICIENTE X^";J-1
60 INPUT P(J)
70 NEXT J
80 INPUT "INTRODUZCA COTA DE LAS RAICES ENTERAS: "
;CO
90 LET X=P(1)
100 DIM D(X):DIM R(X)
110 GOSUB 300
120 FOR J=1 TO L
130 LET K=D(J)
140 GOSUB 400
150 NEXT J
160 IF Z=0 THEN PRINT "NO TIENE RAICES ENTERAS":GO
TO 210
170 PRINT "LAS RAICES ENTERAQS SON: "
180 FOR J=1 TO Z
190 PRINT R(J)
200 NEXT J
210 END
300 REM SUBRUTINA PARA CALCULAR LOS DIVISORES DE X
310 FOR J=1 TO X/2
320 IF R=0 THEN LET D(J)=J:LET L=L+1
340 NEXT J
350 RETURN
400 REM SUBRUTINA PARA APLICAR LA REGLA DE RUFFINI
402 LET S(N-1)=P(N)
405 FOR I=N-1 TO 2 STEP -1
410 LET S(I-1)=P(N)+K*S(I)
420 NEXT I
430 IF S(I)=0 THEN LET R(J)=K:LET Z=Z+1
440 RETURN
```

Hacemos un proceso semejante al de búsqueda de los factores primos, en el sentido de que si un número es raíz del polinomio puede serlo más de una vez. Por ello, cuando detectamos una raíz, «reciclamos» para comprobar si vuelve a repetirse. También verá que si probamos con un valor A, lo hacemos además con −A. Lógico, ya que −A es divisor del coeficiente independiente.

Como aplicación de todo lo dado, sugeriría al lector probar con otras reglas para hallar raíces, por ejemplo, las relaciones de Cardano. Bibliografía de ésta y de otras reglas podrá encontrarse en libros de Cálculo Numérico correspondientes a primero de Carrera. Si no se anima, siga adelante con el libro y no se dé por vencido.

BASES DE NUMERACION 3



EL estudio de los *sistemas de numeración* es uno de los temas más fáciles de comprender y más difíciles de asimilar. La razón de ello es que desde pequeñitos contamos con los dedos de la mano (y de mayores algunos siguen haciéndolo) y, por tanto, nos es familiar contar en un sistema de base 10, pero nos resulta complicado asimilar que 1+1=10 (en base 2).

Con la llegada de los ordenadores se empiezan a poner de moda palabras como el sistema binario (base 2), octal (base 8) y hexadecimal (base 16). Aquí comienzan nuestros problemas, porque el ordenador será muy «hábil» con los 1 y 0, pero nosotros nos manejamos mejor con los números del 0 al 9.

Debido a que nuestro aparato nos ha metido en este embrollo, vamos a pasarle el problema y que él lo resuelva (aunque tendremos que ayudarle).

La base, N, de un sistema de numeración es el número de unidades de un orden que se necesitan para anotar una unidad de orden superior. Así, en base 10, cuando tenemos tres unidades, por ejemplo, escribimos 3, cuando tenemos ocho, escribimos 8 y cuando disponemos de dos veces ocho unidades, es decir, dieciséis unidades, escribimos 16 (10 unidades=1 decena, y 6 unidades más).

De acuerdo con el mismo criterio, en base 6 escribimos el ocho como 12 (un «grupo de 6»=10 y dos unidades más) y en base 3 el mismo ocho lo escribimos como 22 (dos «grupos de 3»=20 y dos unidades más).

Lo primero es conseguir aclarar el paso de cualquier base a base 10 y viceversa. (Vamos a trabajar hasta base 16 por motivos prácticos; si el lector quiere ampliar este tope no conlleva gran dificultad. ¡Inténtelo!)

Empecemos con el *paso de base 10 a cualquier base*. Matemáticamente es ir dividiendo el número dado por la base deseada, quedándonos con los restos, convirtiendo el cociente en el dividendo de la siguiente división.

¿Cuándo acabaría?

Apoyándonos nuevamente en las Matemáticas, sabemos que un número escrito en cualquier base no puede tener un guarismo mayor o igual que la base. Es decir, que si trabajamos en base 4 sólo se puede utilizar el 0, 1, 2 y 3.

Entonces ya tenemos la condición final; no hará falta seguir dividiendo cuando el cociente sea menor que la base. Observe este sencillo programa:

```
10 REM CAMBIO DE BASE 10 A BASE N. POR FCO. MORALES
20 INPUT "INTRODUZCA NUMERO EN BASE 10: ";A
30 INPUT "INTRODUZCA BASE DESEADA: ";B
40 LET C$="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
50 LET M$="":LET A1=A
60 LET C=INT(B*((A/B)-INT(A/B))+1.5)
70 LET M$=MID$(C$,C,1)+M$:REM <-- EN EL SPECTRUM C
AMBIAR ESTA LINEA POR M$=C$(C)+M$
80 LET A=INT(A/B)
90 IF A>0 THEN GOTO 60
100 PRINT "EL NUMERO ";A1;" EN BASE ";B;" ES ";M$
999 END
```

Verá una variable, cifra-$, con una serie de valores del 0 al F. Como dijimos al principio del tema, nos vamos a limitar a 16 posibles bases; por tanto, el valor máximo posible será la F. En base 16, como no disponemos de símbolos para representar los números por encima del 9, se utilizan las letras: A (vale 10), B (valor 11), C, D, E y F (valor 15). Con la función MID$ traduciremos los posibles restos en su valor en la base pedida. Lógicamente hasta la base 10 no haría falta este proceso, pero a partir de ella hay que suplir determinados valores por letra. Si el lector quiere ampliar este margen de bases sólo tendría que añadir letras a CIFRA$.

Para el proceso inverso, *paso de una base cualquiera a base 10,* hay que calcular el valor de cada símbolo, según su posición y valor real, acumulando la suma para el resultado final. Las instrucciones de la 10 a la 30 son iguales al programa anterior. Aquí el control de fin vendrá dado por la cantidad de símbolos que tenga el número, es decir, por la longitud (que calculamos mediante la función LEN del BASIC).

El programa quedaría:

```
10 REM CAMBIO DE BASE N A BASE 10 POR FCO. MORALES
20 INPUT "INTRODUZCA NUMERO EN CUALQUIER BASE: ";N
$
30 INPUT "INTRODUZCA BASE DESEADA: ";B
40 LET C$="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
50 LET L=LEN(N$)
60 LET A=0
70 FOR J=1 TO L
80 FOR I=1 TO B
90 IF MID$(N$,J,1)<>MID$(C$,I,1) THEN GOTO 110:REM
 <-- EN EL SPECTRUM SUSTITUIR ESTA LINEA POR: IF N
$(J)<>C$(I) THEN GOTO 110
100 LET A=A+INT((I-1)*(B^(L-J))+.5)
110 NEXT I
120 NEXT J
130 PRINT "EL NUMERO DESEADO ES: ";A
140 END
```

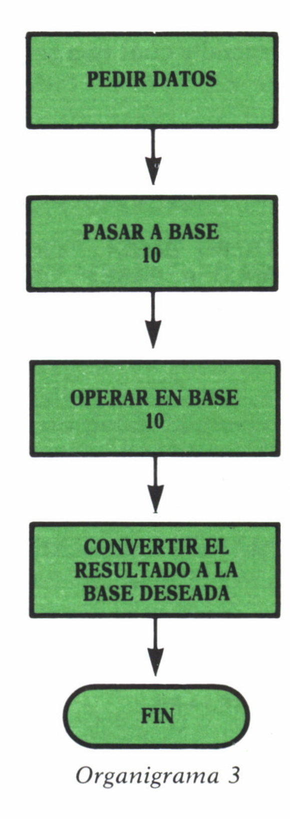
Para el *paso de una base a otra* hay que utilizar como paso intermedio la base 10. El programa consistirá en encadenar los dos anteriores, teniendo en cuenta qué base es la inicial y cuál va a ser la receptora del resultado.

```
10 REM CAMBIO DE UN NUMERO DE UNA BASE B1 A OTRA B 2.
POR FCO. MORALES
20 INPUT "INTRODUZCA NUMERO EN CUALQUIER BASE: ";N
$
30 INPUT "INTRODUCIR SU BASE: ";B1
40 INPUT "PARA PASARLO A BASE: ";B2
50 LET C$="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
60 LET L=LEN(N$)
70 LET A=0
80 GOSUB 220
90 FOR J=1 TO L
100 FOR I=1 TO B1
110 IF MID$(C$,I,1)<>MID$(N$,J,1) THEN GOTO 130:RE
M <-- EN EL SPECTRUM SUSTITUIR ESTA LINEA POR: IF
C$(I)<>N$(J) THEN GOTO 130
120 LET A=A+INT((I-1)*(B1^(L-J))+.5)
130 NEXT I
140 NEXT J
150 LET R$=""
160 LET P=INT(B2*((A/B2)-INT(A/B2))+1.5)
170 LET R$=MID$(C$,P,1)+R$:REM <-- EN EL SPECTRUM
CAMBIARLO POR: R$=C$(P)+R$
180 LET A=INT(A/B2)
190 IF A>0 THEN GOTO 160
200 PRINT "EL NUMERO ";N$;" EN BASE ";B1;" ES IGUAL
A ";R$;" EN BASE ";B2
210 END
220 REM COMPROBACION DE DATOS
```

```
230 FOR J=1 TO L
240 IF MID$(N$,J,1)>MID$(C$,B1,1) THEN PRINT "DATO
 INCORRECTO":PRINT:RUN:REM <-- EN EL SPECTRUM SUST
ITUIR ESTA LINEA POR: IF N$(J)>B$(B1) THEN PRINT "
DATO INCORRECTO":PRINT:RUN
250 NEXT J
260 RETURN
```

Ya le hemos enseñado a nuestro micro a traspasar bases, sigamos educándole diciéndole *cómo se opera en cualquier base.*

El problema consiste en que tampoco sabemos hacerlo nosotros directamente. Para ello nos apoyaremos en lo que sabemos, operar en base 10. Ver organigrama 3.



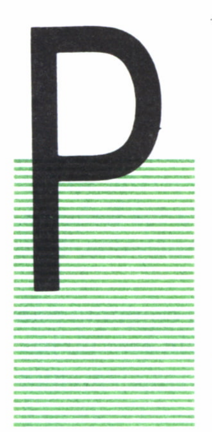
Organigrama 3

Este programa lo sofisticaremos un poco, poniendo controles a los datos. Como indicamos en la introducción, la mayoría de los programas de este libro suponen que los datos son correctos. En este caso, vamos a com-

probar que esto es así y este proceso se podrá utilizar como ejemplo para los demás programas.

```
10 REM OPERACIONES EN DISTINTAS BASES. POR FCO.
MORALES
20 LET C$="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ":M
$=""
30 PRINT "1.- SUMAR DOS NUMEROS (A+B)
40 PRINT "2.- RESTAR DOS NUMEROS (A-B)
50 PRINT "3.- MULTIPLICAR DOS NUMEROS (A*B)
60 PRINT "4.- DIVIDIR DOS NUMEROS (A/B)
70 PRINT
80 PRINT "PULSE EL NUMERO DE LA OPCION DESEADA"
90 INPUT N
100 PRINT
110 INPUT "DEME LA BASE EN LA QUE OPERAREMOS: ";B
120 PRINT
130 INPUT "DEME EL PRIMER NUMERO: ";N$
140 PRINT:GOSUB 290:LET N1=A
150 INPUT "DEME EL SEGUNDO NUMERO: ";N$
160 PRINT:GOSUB 290:LET N2=A
170 IF N=1 THEN LET R=N1+N2
180 IF N=2 THEN LET R=N1-N2:IF R<0 THEN LET R=N2-N
1
190 IF N=3 THEN LET R=N1*N2
200 IF N=4 THEN LET R=INT(N1/N2)
210 LET R1=R
220 LET C=INT(B*((R/B)-INT(R/B))+1.5)
230 LET M$=MID$(C$,C,1)+M$:REM <-- EN EL SPECTRUM
SUSTITUIR ESTA LINEA POR: M$=C$(C)+M$
240 LET R=INT(R/B)
250 IF R>0 THEN 220
260 PRINT "EL RESULTADO PEDIDO ES ";M$;" EN BASE "
;B;
270 PRINT " Y ";R1;" EN BASE 10"
280 END
290 REM PASO A BASE 10
300 LET L=LEN(N$)
310 LET A=0
320 FOR J=1 TO L
330 FOR I=1 TO B
340 IF MID$(C$,I,1)<>MID$(N$,J,1) THEN GOTO 360:RE
M <-- EN EL SPECTRUM SUSTITUIR ESTA LINEA POR: IF
C$(I)<>N$(J) THEN GOTO 360
350 LET A=A+INT((I-1)*(B^(L-J))+.5)
360 NEXT I
370 NEXT J
380 RETURN
```

CONJUNTOS 4



POR medio de los conjuntos se va a realizar una serie de programas, cuya aplicación directa quizá no sea práctica, pero valdrán para procesos posteriores. Así, por medio de la relación de pertenencia, se tocará el tema de la búsqueda, algo muy útil en el tratamiento de datos para seguir después con las relaciones de inyección, biyección, etc., por medio de las tablas; esto servirá, al lector interesado, para el estudio de grafos y circuitos. Terminaremos con un juego, el MASTER MIND, como aplicación de lo visto a lo largo del capítulo.



¿QUE ES UN CONJUNTO?

Como todos sabemos, no es más que una colección de elementos. Hemos de reseñar que para simplificar los programas, se tratarán conjuntos numéricos. Queda a gusto del lector probar con otro tipo de datos.

Empecemos con la *relación de pertenencia*, sabiendo que un elemento pertenece a un conjunto si está en él. Por tanto, no hay más que buscarlo en dicho conjunto. Observe el programa:

```
10 INPUT "INTRODUZCA NUMERO DE ELEMENTOS DE A: ";N
20 DIM A(N)
30 PRINT "INTRODUZCA ELEMENTOS DE A:"
40 FOR J=1 TO N
50 INPUT A(J)
60 NEXT J
```

```
70 INPUT "INTRODUZCA DATO A BUSCAR: ";B
80 FOR J=1 TO N
90 IF A(J)=B THEN PRINT B;" ESTA EN A":GOTO 999
100 NEXT J
110 PRINT B;" NO ESTA EN A"
999 END
```

Lógicamente lo primero a efectuar es la entrada al ordenador de los elementos pertenecientes al conjunto.

Después observará que aparte de pedir el dato buscado, se produce un proceso reiterativo de lectura-comparación. De esta forma, se lee uno a uno cada elemento del conjunto (Tabla), buscando el deseado. Si se encuentra, ya se sabe que pertenece; en caso contrario, si al recorrer toda la tabla no hay éxito, la respuesta sería negativa.

Este proceso de búsqueda de un elemento es aplicable a los ficheros. Como información al lector más avanzado, existen varios métodos. Esto se justifica al intentar ganar tiempo en estas búsquedas *.

Procedamos ahora a analizar el concepto de *inclusión*. Un conjunto está incluido en otro si todos sus elementos están en el otro. Según esto no hay más que repetir el proceso de búsqueda para cada uno de los elementos del conjunto.

Fíjese en el programa siguiente:

```
10 REM BUSCA SI B ES SUBCONJUNTO DE A POR FCO.
MORALES
20 INPUT "INTRODUZCA NUMERO DE ELEMENTOS DE A: ";N
30 DIM A(N)
40 PRINT "INTRODUZCA ELEMENTOS DE A:"
50 FOR J=1 TO N
60 INPUT A(J)
70 NEXT J
80 INPUT "INTRODUZCA DIMENSION DE B: ";M
90 DIM B(M)
100 PRINT "INTRODUZCA ELEMENTOS DE B: "
110 FOR J=1 TO M
120 INPUT B(J)
130 NEXT J:LET X=0
140 FOR I=1 TO N
150 FOR J=1 TO M
```

* N. del A.: El estudio de los diversos métodos y la elección para cada tipo de estructura es algo muy estudiado por los expertos. Recomendaría, a los interesados en el tema, la lectura del libro de Algoritmos+Datos=PROGRAMA, en donde encontrará información sobre ello.

```
160 IF B(J)=A(I) THEN LET X=X+1
170 NEXT J:NEXT I
180 IF X>=M THEN PRINT "B ESTA INCLUIDO EN A":GOTO
 200
190 PRINT "B NO ESTA INCLUIDO EN A"
200 END
```

Como verá, no tiene gran complicación este programa.

Estudiemos ahora las operaciones de la unión y la intersección. Para ello se necesita otra tabla que sea el resultado.

En el caso de la *intersección* hay que buscar los elementos comunes. Aprovechando el programa de la inclusión, se obtendrá fácilmente la solución.

La intersección será repetir la operación de búsqueda de la inclusión, con la diferencia ahora de que se irán colocando en una tercera tabla los elementos comunes. Esta será la solución:

```
10 INPUT "INTRODUZCA NUMERO DE ELEMENTOS DE A: ";N
20 LET Z=0:DIM A(N)
30 PRINT "INTRODUZCA ELEMENTOS DE A:"
40 FOR J=1 TO N
50 INPUT A(J)
60 NEXT J
70 INPUT "INTRODUZCA DIMENSION DE B: ";M
80 DIM B(M)
90 PRINT "INTRODUZCA ELEMENTOS DE B:"
100 FOR J=1 TO M
110 INPUT B(J)
120 NEXT J
130 LET X=M:IF N>M THEN LET X=N
140 DIM C(X)
150 FOR J=1 TO M
160 FOR I=1 TO N
170 IF A(I)=B(J) THEN LET Z=Z+1:LET C(Z)=A(I)
180 NEXT I
190 NEXT J
200 PRINT "LA INTERSECCION DE A Y B ES:"
210 FOR I=1 TO Z
220 PRINT C(I)
230 NEXT I
999 END
```

La *unión* la apoyaremos en la intersección. El resultado de la unión se almacenará en una tabla cuya dimensión máxima será (N+M). El camino

más fácil consistirá en copiar directamente los elementos de uno de los conjuntos a la matriz solución. Posteriormente, con el otro se efectúa el proceso de búsqueda para cada uno de sus elementos. Si se halla, quiere decir que pertenece a la intersección; por tanto, no es necesario repetirlo en unión. En caso contrario, se transportan a la tabla unión creada.

```
10 INPUT "INTRODUZCA NUMERO DE ELEMENTOS DE A: ";N
20 DIM A(N):LET Z=0
30 PRINT "INTRODUZCA ELEMENTOS DE A:"
40 FOR J=1 TO N
50 INPUT A(J)
60 NEXT J
70 INPUT "INTRODUZCA DIMENSION DE B: ";M
80 DIM B(M)
90 PRINT "INTRODUZCA ELEMENTOS DE B:"
100 FOR J=1 TO M
110 INPUT B(J)
120 NEXT J
130 DIM S(N+M)
140 FOR J=1 TO M
150 FOR I=1 TO N
160 IF B(J)=A(I) THEN GOTO 190
170 NEXT I
180 LET Z=Z+1:LET S(N+Z)=B(J)
190 NEXT J
200 FOR I=1 TO N
210 LET S(I)=A(I)
220 NEXT I
230 PRINT "LA UNION DE A Y B ES:"
240 FOR J=1 TO N+Z
250 PRINT S(J)
260 NEXT J
999 END
```

Ya se han hecho las operaciones básicas de la teoría de conjuntos. Antes de seguir con el producto cartesiano y las relaciones de elementos, vamos a crear un programa «sort».

Un *sort* es un proceso conducente a la ordenación de datos, ya sea de un conjunto, un fichero, etc. Aquí nos vamos a centrar en un conjunto (por lógica realmente estamos haciendo la clasificación de una tabla).

El método más simple, y más largo, es el de la «baraja», así llamado por su semejanza al proceso realizado cuando se ordena una baraja de naipes.

Si nosotros cogemos un mazo de cartas para su ordenación, podremos ver solamente dos a la vez. Es decir, en un momento determinado se com-

para por pares. Fruto de esta comparación, nos quedaremos con la más alta (vamos a ordenar de mayor a menor). Una vez elegida la mayor, comparamos ésta con la siguiente, repitiéndose el proceso hasta terminar la baraja.

De esta manera, al final de una pasada completa, está colocada la mayor en primer lugar. Repitiendo todo el proceso para conseguir la segunda más alta; para ello no empezamos desde la primera, sino ya desde la segunda. Se reiteraría el proceso N−1 veces. El proceso sería el siguiente:

```
10 INPUT "INTRODUZCA NUMERO DE ELEMENTOS DE A: ";N
20 DIM A(N)
30 PRINT "INTRODUZCA ELEMENTOS DE A:"
40 FOR J=1 TO N
50 INPUT A(J)
60 NEXT J
70 FOR K=1 TO N
80 FOR J=1 TO K
90 IF A(K)<A(J) THEN LET X=A(K):LET A(K)=A(J):LET
A(J)=X
100 NEXT J
110 NEXT K
120 PRINT "EL SUBCONJUNTO ORDENADO QUEDA:"
130 FOR I=1 TO N
140 PRINT A(I)
150 NEXT I
999 END
```

Desde la instrucción 60 a la 100, se produce el intercambio. Si la carta mayor hasta ese momento es mayor o igual que la leída, sigue conservando su trono. El caso de cartas iguales quedaría de esta forma solucionado, ya que sería mayor la que antes se leyera.

Para obtener el producto cartesiano de dos conjuntos se define una matriz cuadrada A(N,N), de tal forma que diremos que si dos elementos I y J están relacionados, el valor A(I,J) es 1; en caso contrario es nulo.

Este procedimiento de operación es muy útil a la hora de operar con la teoría de grafos y en el estudio de circuitos por medio del ordenador. Veremos básicamente cómo jugar con ello, pero a los interesados aconsejaría profundizar en el tema.

El siguiente programa analizará si *una relación es de equivalencia*. Para ello deberá ser: simétrica, reflexiva y transitiva. Estas tres condiciones son imprescindibles; en el momento que una de ellas no se cumpliera no haría falta seguir. Nuestro micro no sabe nada de conjuntos y como siempre

tendremos que proporcionarle el algoritmo correspondiente. Observe el programa:

```
10 INPUT "INTRODUZCA DIMENSION DE C: ";N
20 DIM C(N,N)
30 PRINT "INTRODUZCA LOS PARES DE ELEMENTOS RELACCIONADOS A,B :"
40 INPUT A,B
45 LET C(A,B)=1
50 INPUT "MAS DATOS (S/N) ";A$
60 IF A$="SI" THEN GOTO 40
70 FOR J=1 TO N
80 IF C(J,J)<>1 THEN PRINT "NO CUMPLE LA PRIPIEDAD REFLEXIVA":GOTO 999
85 NEXT J
90 FOR J=1 TO N
100 FOR I=1 TO N
110 IF C(J,I)<>C(I,J) THEN PRINT "NO CUMPLE LA SIMETRICA":GOTO 999
120 NEXT I
130 NEXT J
140 FOR J=1 TO N
150 FOR I=1 TO N
160 FOR K=1 TO N
170 IF A(J,I)=1 AND A(I,K)=1 AND A(J,K)=0 THEN PRINT "NO CUMPLE LA TRANSITIVA":G
OTO 999
180 NEXT K
190 NEXT I
200 NEXT J
999 END
```

Entendemos que está bastante claro el proceso. Se ve claramente que en el momento que un elemento no cumple una propiedad, el conjunto no la cumple y, por tanto, la relación de equivalencia no se produce. Pruebe con otro tipo de relaciones.

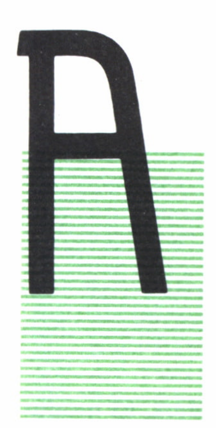
Dejando las teorías, juguemos un poco aplicando lo visto anteriormente. Enseñemos a nuestro ordenador el **MASTER MIND** o más conocido como el juego de los números o de los colores. Consiste en introducir al ordenador una combinación de cinco números, y otro jugador tendrá que ir probando hasta averiguarla. El ordenador le irá diciendo los muertos y los heridos (aciertos en color y posición o aciertos sólo en color sin que coincida la posición indicada para dicho color).

```
10 DIM A(5),B(5)
20 PRINT "INTRODUZCA LA COMBINACION SECRETA"
30 FOR J=1 TO 5
40 INPUT A(J)
50 NEXT J
60 CLS :REM PARA COMMODORE CAMBIAR POR PRINT "[SHI
FT-HOME]"
70 PRINT "INTRODUZCA COMBINACION"
80 FOR J=1 TO 5
90 INPUT B(J)
100 NEXT J
```

```
110 LEM M=0:LET H=0
120 FOR K=1 TO 5
130 IF A(K)=B(K) THEN M=M+1
140 NEXT K
150 FOR J=1 TO 5
160 FOR L=1 TO 5
170 IF L=J THEN GOTO 190
180 IF A(J)=B(L) THEN LET H=H+1:GOTO 200
190 NEXT L
200 NEXT J
210 PRINT "MUERTOS: ";M;" HERIDOS: ";H
220 IF M=5 THEN PRINT "LO LOGRO":GOTO 999
230 LET M=0:LET H=0:GOTO 70
999 END
```

Las condiciones adicionales que se han impuesto en el programa son: que se juegue con números y que no valga repetir. Si se atreve, pruebe con colores (sólo tiene que definir las tablas alfanuméricas) y repitiendo colores.

COMPLEJOS 5



pesar de lo que su nombre pueda sugerir, realmente los complejos no son tan «complejos», sino que las operaciones necesarias para su manejo predisponen al equívoco. En sumas y restas lo más fácil es la forma binómica; en cambio, en productos, divisiones y exponenciales lo más correcto es la forma polar. El trasiego de operaciones intermedias para trasvasar de una forma a otra origina, muchas veces, fallos. Todo aquel que haya trabajado con complejos, sobre todo en su aplicación a circuitos de corriente alterna, nos comprenderá.

Se intentará evitar todo este trasiego consiguiendo que nuestro ordenador trate adecuadamente el tema. Nos encontraremos con que nuestro equipo comienza a poner pegas. La primera consiste en el uso de las funciones trigonométricas; él trabaja en radiones, nosotros más comúnmente en grados. Existe un comando en BASIC llamado DEG, que se encarga de la misión de *pasar de una unidad de medida a la otra.* Pero puede ocurrir que un determinado modelo no lo tenga (¡no todos son tan listos!). Para ello, facilitémosle su tarea, creando unas subrutinas que se encarguen de esta tarea.

```
10 REM PASO DE GRADOS A RADIANES
20 INPUT "INTRODUCIR GRADOS: ";G
30 LET R=G*2*3.14159/360
40 PRINT "EN RADIANES ES ";R
```

```
10 REM PASO DE RADIANES A GRADOS
20 INPUT "INTRODUCIR RADIANES: ";R
```

```
30 LET G=R*360/(2*3.14159)
40 PRINT "EN GRADOS ES: ";G
```

La base de esas subrutinas es una simple regla de tres, teniendo como datos fijos 2π y 360°.

Ya no tendrá problema con los ángulos; ahora pasemos a operar con complejos. Hemos dicho que necesitaremos *pasar de una forma binómica a polar y viceversa*. Programémosle primero para realizar el paso de binómica a polar, utilizando nuestra rutina de traspaso radiones-grados.

```
10 REM PASO DE BINOMICA A POLAR
20 INPUT "INTRODUZCA DATOS (A,B): ";A,B
30 LET M=SQR(A^2+B^2)
40 LET F=ATN(A/B)
50 LET G=F*360/(2*3.14159)
60 PRINT "EL COMPLEJO EN FORMA POLAR ES: ";M;"/";F
```

Se han aplicado las fórmulas de paso de números complejos. El módulo (30) y la fase (40) se calculan directamente a partir de A y B. El proceso inverso es claro, simplemente hay que invertir los papeles según estas dos fórmulas:

$$A = M \cos\alpha$$
$$B = M \operatorname{Sen}\alpha$$

```
10 REM PASO DE POLAR A BINOMICA
20 INPUT "INTRODUZCA LOS DATOS (MODULO,FASE): ";M,
F
30 LET A=M*COS(F)
40 LET B=M*SIN(F)
50 PRINT "EL BINOMIO RESULTANTE ES: ";A;",";B
```

Estamos utilizando la función ATN (arcotangente); si en su modelo no existe, debe haber una similar; sustituya ATN por su función.

Ya podemos hacer todas las transformaciones necesarias. Indicamos al principio que para la *suma y resta de quebrados* la forma binómica era la más adecuada. Observe cómo se hace, ya que el complejo suma (resta) no es más que la suma (resta) de los respectivos términos de los dos complejos a sumar (restar).

```
10 REM SUMA Y DIFERENCIA EN FORMA BINOMICA
20 INPUT "INTRODUZCA PRIMER OPERANDO EN FORMA BINO
MICA: ";A1,B1
30 INPUT "INTRODUZCA SEGUNDO OPERANDO EN FORMA BIN
OMICA: ";A2,B2
40 PRINT "LA SUMA DE LOS DOS COMPLEJOS ES: ";A1+A2
;",";B1+B2
50 PRINT "LA RESTA DE LOS DOS COMPLEJOS ES: ";A1-A
2;",";B1-B2
```

Sencillo, ¿no cree?
Sigamos con la *multiplicación, división y exponenciación:*

```
10 REM MULTIPLICACION, DIVISION Y EXPONENCIACION
20 INPUT "INTRODUZCA PRIMER OPERANDO EN FORMA POLA
R: ";M1,F1
30 INPUT "INTRODUZCA SEGUNDO OPERANDO EN FORMA POL
AR: ";M2,F2
40 PRINT "LA MULTIPLICACION ES: ";M1*M2;",";F1*F2
50 PRINT "LA DIVISION ES: ";M1/M2;",";F1/F2
60 INPUT "INTRODUZCA EXPONENTE: ";C
70 PRINT "EL PRIMER OPERANDO ELEVADO A ";C; "ES ";
M1^C;"/";F1^C
```

Tampoco tiene problemas porque en forma polar es muy fácil realizar estas operaciones, utilizando las siguientes fórmulas:

$$M1\underline{|\alpha} * M2\underline{|\beta} = M1 * M2\underline{|\alpha+\beta}$$
$$M1\underline{|\alpha} / M2\underline{|\beta} = M1/M2\underline{|\alpha-\beta}$$
$$(M1\underline{|\alpha})^{\times} = M1^{\times}\underline{|\times\alpha}$$

Ya tenemos todo lo necesario para crear nuestro superprograma que *opere cualquier número complejo,* venga en la forma que sea. Para ello bifurcaremos a una determinada dirección dependiendo de la operación a realizar, analizando la forma de los operandos, transformándola en la más adecuada para operar.

```
10 REM OPERACIONES CON NUMEROS COMPLEJOS
20 INPUT "INTRODUZCA FORMA DE LOS DATOS (B=BINOMIC
A,P=POLAR): ";A$
```

```
30 INPUT "INTRODUZCA PRIMER OPERANDO: ";A1,B1
40 INPUT "INTRODUZCA SEGUNDO OPERANDO: ";A2,B2
50 PRINT "ELIJA OPERACION:":PRINT
60 PRINT "1.- SUMA"
70 PRINT "2.- RESTA"
80 PRINT "3.- MULTIPLICACION"
90 PRINT "4.- DIVISION"
100 INPUT O
110 IF O=1 THEN GOTO 200
112 IF O=2 THEN GOTO 300
114 IF O=3 THEN GOTO 400
116 IF O=4 THEN GOTO 500
120 END
200 IF A$="P" THEN GOSUB 600
210 LET AS=A1+A2
220 LET BS=B1+B2
230 PRINT "LA SUMA EN FORMA BINOMICA ES: ";AS,BS
240 GOTO 999
300 IF A$="P" THEN GOSUB 600
310 LET AS=A1-A2
320 LET BS=B1-B2
330 PRINT "LA RESTA EN FORMA BINOMICA ES: ";AS,BS
340 GOTO 999
400 IF A$="B" THEN GOSUB 700
410 LET AS=A1*A2
420 LET BS=B1*B2
430 PRINT "LA MULTIPLICACION EN FORMA POLAR ES: ";
AS,BS
440 GOTO 999
500 IF A$="B" THEN GOSUB 700
510 LET AS=A1/A2
520 LET BS=B1/B2
530 PRINT "LA DIVISION EN FORMA POLAR ES: ";AS,BS
540 GOTO 999
600 LET M1=A1:LET M2=A2
610 LET A1=M1*COS(B1):LET B1=M1*SIN(B1)
620 LET A2=M2*COS(B2):LET B2=M2*SIN(B2)
630 RETURN
700 LET X1=A1:LET X2=A2
710 LET A1=SQR(X1^2+B1^2)
720 LET A2=SQR(X2^2+B2^2)
730 LET B1=ATN(X1/B1)
740 LET B2=ATN(X2/B2)
750 RETURN
999 END
```

Hemos dejado para lo último la *radicación de un complejo*. El motivo es que aparece un fenómeno curioso en esta radicación. Con complejos,

la raíz N da lugar a N raíces, separadas 360°/N una de otra. El programa calcula el valor de estas raíces hasta un N; fíjese cómo el módulo y el ángulo de fase disminuyen, ya que

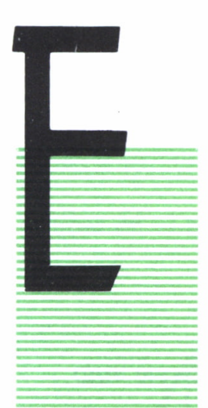
$$\sqrt[N]{(M\angle\alpha)} = \sqrt[N]{M}\angle\alpha/N.$$

```
10 REM RADICACION DE UN NUMERO COMPLEJO
20 INPUT "INTRODUZCA COMPLEJO EN FORMA POLAR: ";M,
F
30 INPUT "INTRODUZCA NUMERO TOPE DE RADICACION: ";
N
40 PRINT "MODULO","DESFASE"
50 PRINT "======","======="
60 FOR J=1 TO N
70 PRINT M^(1/J),F/J
80 NEXT J
```

```
INTRODUZCA COMPLEJO EN FORMA POLAR:  128  50
INTRODUZCA NUMERO TOPE DE RADICACION:  10
MODULO                       DESFASE
======                       =======
 128            50
 11.313708498983              25
 5.03968419957 86             16.666666666667
 3.3635856610146              12.5
 2.6390158215456              10
 2.2449240966186              8.3333333333333
 1.9999999999998              7.1428571428571
 1.8340080864092              6.25
 1.7144879657061              5.5555555555556
 1.6245047927123              5
```

Este programa gráficamente es bastante interesante. Como las instrucciones gráficas de un micro a otro varían, vamos a indicarle cómo hacerlo: pruébelo. Según aumenta N, las circunferencias formadas por las raíces van siendo más pequeñas y los radios que unen el centro con las raíces se juntan. Dibuje una circunferencia por cada valor del radio ($R=\sqrt[n]{\text{módulo}}$), y coloque los puntos de las raíces; tenga en cuenta que tendrán que ir girando 360°/N. Pruébelo, le gustará como queda.

POLIGONOS 6



ESTAMOS consiguiendo progresos en nuestro micro; intentemos ahora introducirle las fórmulas de geometría para descargar nuestra memoria, ya que la suya funciona mejor. No por capacidad; la capacidad del hombre no es comparable a ninguno de los grandes monstruos de ordenadores, sino por estar más «despejada».

Juguemos primero con los pequeños polígonos, *cuadrados, trapecios, rombos y rectángulos* (los triángulos los trataremos en el siguiente capítulo).

Hagamos que el ordenador nos vaya interrogando hasta tener los datos necesarios para el cálculo del área. Esta se calculará aplicando las superconocidas fórmulas geométricas respectivas; si no las recuerda en este momento, las encontrará en las líneas 110, 210, 310 y 410.

```
10 REM CALCULO DE AREAS DE CUADRILATEROS
20 PRINT "ELIJA OPCION"
30 PRINT "1.- CUADRADO"
40 PRINT "2.- RECTANGULO"
50 PRINT "3.- ROMBO"
60 PRINT "4.- TRAPECIO"
70 INPUT O
80 IF O<1 OR O>3 THEN RUN
90 IF O=1 THEN INPUT "INTRODUZCA LADO: ";L:LET A=L
*L
100 IF O=2 OR O=3 THEN INPUT "INTRODUZCA LADOS: ";
A,B:LET A=A*B
110 IF O=4 THEN INPUT "INTRODUZCA DIAGONALES: ";D1
,D2:LET A=D1*D2/2
120 PRINT "EL VALOR DEL AREA ES: ";A
999 END
```

Vamos a «jugar» ahora con *el círculo*. En él se dan conceptos (segmento, corona, etc.) no tan habituales y utiliza fórmulas en el programa para que nuestro ordenador aclare la situación. Vamos a emplear un «MENU»: que es un buen método para elegir opciones. También se utilizan las instrucciones ON GOSUB y ON GOTO para bifurcar según la opción tecleada. El ordenador nos irá pidiendo los datos necesarios a medida que los necesite; al igual que en el anterior programa, encontrará las fórmulas aplicadas en el propio programa.

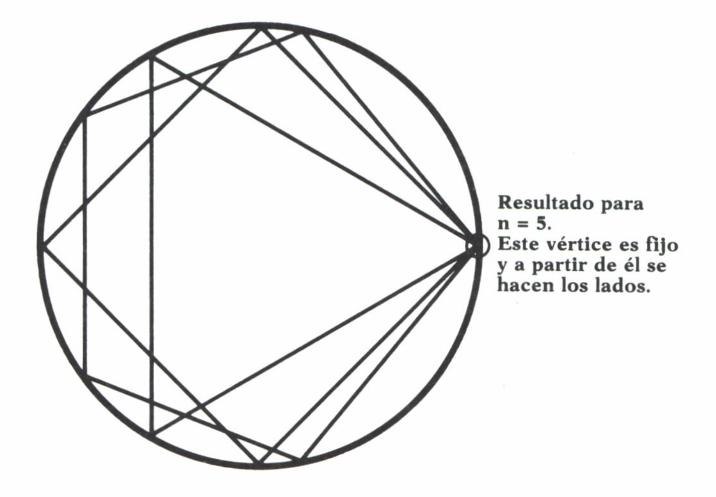
```
10 REM CALCULO DE AREAS DE UN CIRCULO
15 LET PI=3.14159
20 PRINT "ELIJA OPCION"
30 PRINT "1.- CIRCULO"
40 PRINT "2.- SECTOR CIRCULAR"
50 PRINT "3.- SEGMENTO CIRCULAR"
60 PRINT "4.- TRAPECIO CIRCULAR"
70 PRINT "5.- CORONA CIRCULAR"
80 INPUT O
85 IF O<1 OR O>5 THEN RUN
90 PRINT:INPUT "RADIO: ";R
100 IF O=1 THEN LET A=PI*R^2
110 IF O=2 THEN INPUT "INTRODUZCA EL ANGULO EN RAD
IANES: ";AN:LET A=AN*R^2/2
120 IF O=3 THEN INPUT "INTRODUZCA EL ANGULO EN RAD
IANES: ";AN:LET A=(AN-SIN(AN))*R^2/2
130 IF O=4 THEN INPUT "INTRODUZCA EL RADIO MENOR:
";R1:INPUT "INTRODUZCA EL ANGULO EN RADIANES: ";AN
:LET A=AN*(R^2-R1^2)/2
140 IF O=5 THEN INPUT "INTRODUZCA EL RADIO MENOR:
";R1:LET A=(R^2-R1^2)*PI
150 PRINT "EL VALOR DEL AREA ES: ";A
999 END
```

Como se observará, se ha utilizado la técnica de ir preguntando al usuario del programa por los datos necesarios, en vez de exigirle que los introduzca. Esta técnica es muy útil cuando van a utilizar el programa personas que no conocen el ordenador ni el proceso seguido en su programación.

A continuación vamos a ver cómo calcular el área de un polígono tomando como dato el radio de la circunferencia circunscrita. Si mantenemos fijo éste y aumentamos el número de lados, resulta que el área del polígono se acerca al de la circunferencia. Si incrementamos mucho el número de lados llegará un momento en que el polígono y la circunferencia se confundan. Veamos esto con el siguiente programa. Imprimimos cada N veces para no hacer muy grande la solución.

```
10 REM CALCULO DEL AREA DE UN POLINOMIO CIRCUNSCRI
TO
20 INPUT "INTRODUZCA VALOR DE N: ";N
25 INPUT "RADIO: ";R
30 PRINT "LADOS","AREA"
40 PRINT "=====","===="
50 FOR J=3 TO N
55 LET F=2*3.14159/J
60 LET B=2*R*SIN(F/2)
70 LET H=SQR(R^2-(B/2)^2)
80 LET A=J*B*H/2
90 PRINT J,A
100 NEXT J
110 PRINT "EL AREA DEL CIRCULO ES: ";3.14159*R^2
999 END
```

Fíjese en el resultado obtenido por nuestro programa.



Resultado para n = 5. Este vértice es fijo y a partir de él se hacen los lados.

```
INTRODUZCA VALOR DE N:  40
RADIO:  3
LADOS          AREA
=====          ====
 3              11.691354892223
 4              17.999999999984
 5              21.39876423659
 6              23.382673961017
 7              24.627676807361
```

```
             8                25.455827235369
             9                26.032879897392
             10               26.450317031966
             11               26.761700372974
             12               26.999979317312
             13               27.186284417824
             14               27.33465404719
             15               27.454701590042
             16               27.55318506591
             17               27.634965193729
             18               27.703609167352
             19               27.761782028703
             20               27.811506780321
             21               27.854341160547
             22               27.891500212395
             23               27.923942818065
             24               27.952433802536
             25               27.977589174046
             26               27.999909533315
             27               28.01980505665
             28               28.037614394968
             29               28.053619123585
             30               28.068054899976
             31               28.081120158934
             32               28.092982946908
             33               28.103786336796
             34               28.113652750271
             35               28.122687432276
             36               28.130981262563
             37               28.138613044806
             38               28.145651381422
             39               28.152156217498
             40               28.158180118964
            EL AREA DEL CIRCULO ES:   28.27431
```

Esta aproximación se nota más gráficamente. Como sucedía con los complejos, el procedimiento es dibujar la circunferencia, y colocar los vértices girando con el ángulo 360°/N; después unir dos a dos éstos. El programa resultante lo encontrará en el programa siguiente:

```
10 REM GRAFICA DE POLIGONOS CIRCUNSCRITOS POR FCO.
 MORALES
20 CLS:REM EN EL COMMODORE PONER PRINT "[SHIFT-HOM
E]"
```

```
30 INPUT "INTRODUZCA VALOR INICIAL: ";T1
40 INPUT "INTRODUZCA VALOR TOPE: ";T2
50 SCREEN 2:REM 'SCREEN 2'<-- VALIDO PARA M.S.X.,
PARA EL I.B.M. CAMBIARLO POR 'SCREEN 1', Y QUITARL
O PARA SPECTRUM Y AMSTRAD
60 FOR J=0 TO 2*3.14159 STEP .07
70 PSET (70+70*COS(J),70+70*SIN(J)):REM 'PSET'<--
VALIDO PARA M.S.X. e I.B.M., PARA LOS DEMAS CAMBIA
R EL 'PSET' POR 'PLOT'
80 NEXT J
90 FOR I=T1 TO T2
100 LET A=70+70*COS(0):LET B=70+70*SIN(0)
110 PSET (A,B):REM 'PSET'<-- VALIDO PARA M.S.X. e
I.B.M., PARA LOS DEMAS CAMBIAR EL 'PSET' POR 'PLOT
'
120 FOR J=1 TO I
130 LET X=70+70*COS(6.2832/I*J)
140 LET Y=70+70*SIN(6.2832/I*J)
150 LINE -STEP (X-A,Y-B):REM 'LINE -STEP'<-- VALID
O PARA M.S.X e I.B.M., EN EL AMSTRAD SUSTITUIR POR
 'DRAWR' Y EN EL SPECTRUM POR 'DRAW'
160 LET A=X
170 LET B=Y
180 NEXT J
190 NEXT I
200 GOTO 200
```

TRIANGULOS 7

EN el capítulo anterior vimos la manera de calcular las áreas de los distintos polígonos. No abordamos en él los triángulos, debido al tratamiento especial que le vamos a dar a este tema. Esta figura es la más estudiada por los matemáticos, y base de los cálculos realizados en las demás. Trataremos las funciones trigonométricas; éstas representan relaciones existentes entre los lados y ángulos de un triángulo.

Empecemos creándonos nuestra propia subrutina para el cálculo del *seno*. Nos basaremos en el desarrollo de Taylor:

$$\mathrm{Sen}(x)=X-\frac{X^5}{5!}+\frac{X^7}{7!}+(-1)^{n-1}*\frac{X^{(2n+1)}}{2n+1!}$$

En esta expresión obtendremos mejor o peor aproximación dependiendo de la cantidad de términos que tomemos; cuanto mayor sea ésta, más exacto será nuestro cálculo. Para nuestro proceso necesitamos primero calcular el factorial de un número; lo veremos levemente, ya que en el capítulo 12 lo trataremos con profundidad.

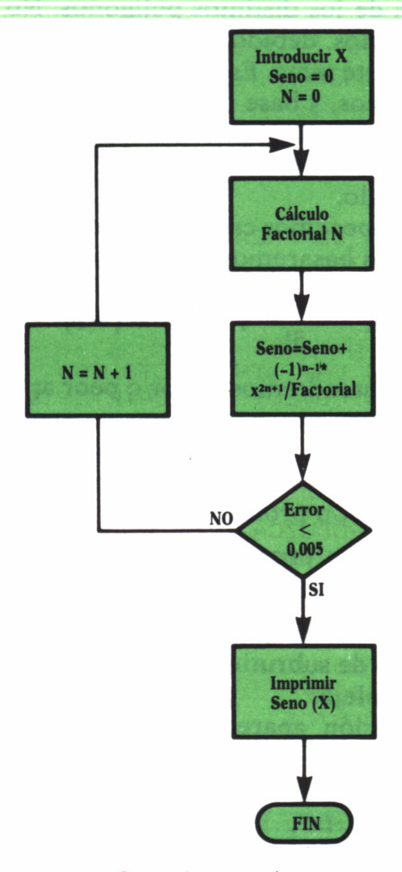
Lo interesante de este programa, a partir del cálculo del seno, es:

1.º Conocer el factorial de un número.
2.º El manejo de un proceso reiterativo controlando el final.
3.º La creación de subrutinas que faciliten la resolución de un problema complejo.

A continuación aparecen el programa y el organigrama 4.

```
10 REM CALCULO DEL VALOR DEL SENO
20 INPUT "INTRODUZCA VALOR DE X: ";X
```

```
30 LET SN=0
40 LET N=0
45 LET AN=0
50 GOSUB 300
60 LET SN=SN+((-1)^(N)*(X^(2*N+1))/FA)
70 IF ABS(SN-AN)<5E-05 THEN PRINT "EL VALOR BUSCAD
O ES ";SN:GOTO 100
80 LET N=N+1:LET AN=SN
90 GOTO 50
100 END
300 REM CALCULO DEL FACTORIAL
310 LET FA=1:IF N=0 THEN RETURN
320 FOR J=1 TO N*2+1
330 LET FA=FA*J
340 NEXT J
350 RETURN
```



Organigrama 4

Como observará, el programa se compone de un proceso reiterativo basado en Seno=Seno+$(-1)^{n-1}$ * X^{2n+1}/FACTORIAL, siendo el segundo seno el acumulado hasta el momento. El error vendrá dado por la diferencia entre el hallado y el calculado anteriormente. Si esta diferencia es mínima ($<0,005$), ya no es necesario seguir. La subrutina factorial (300-360) se estudiará más adelante.

Comenzaremos con el *triángulo, rectángulo* dada su simplicidad, gracias al ángulo recto que tiene. Existen cinco posibles incógnitas, dos ángulos y los tres lados; necesitamos dos para conocer las demás, una de éstas debe ser un lado. La razón es lógica, conocido los tres ángulos sólo podemos saber la proporción entre los tres lados; haría falta un lado para calcular los otros dos. Haremos que nuestro micro nos vaya pidiendo los datos iniciales, él después hará los cálculos. Existe una primera parte (10 a la 96) en la que se interrogará sobre la clase de datos disponibles, bifurcando, según éstos, a la dirección de programa adecuada. En los diferentes puntos a los que se ha bifurcado se pedirán los datos a aplicar en las fórmulas. Como en todos los programas de geometría, los procesos de cálculo no tienen dificultad alguna, pues se limitan a incluir en la línea de programa correspondiente la fórmula matemática que corresponda aplicar.

```
10 REM CALCULO DE AREA PARA TRIANGULO RECTANGULO
20 INPUT "CONOZCO LA HIPOTENUSA(S/N): ";S$
30 IF S$="S" THEN LET HI=1
40 INPUT "CONOZCO UN CATETO(S/N): ";S$
50 IF S$="S" THEN LET CA=1
60 IF HI=1 AND CA=1 THEN GOTO 100
70 INPUT "CONOZCO LOS DOS CATETOS(S/N): ";S$
80 IF S$="S" THEN GOTO 200
90 INPUT "CONOZCO UN LADO Y UN ANGULO(S/N): ";S$
95 IF S$="S" THEN GOTO 300
96 PRINT "NO HAY DATOS SUFICIENTES":GOTO 999
100 INPUT "INTRODUZCA HIPOTENUSA Y CATETO: ";A,B
110 LET C=SQR(A^2-B^2)
120 LET AA=ATN(B/C)
130 LET AB=ATN(C/B)
140 GOTO 400
200 INPUT "INTRODUZCA LOS DOS CATETOS: ";B,C
210 LET A=SQR(B^2+C^2)
220 LET AA=ATN(B/C)
230 LET AB=ATN(C/B)
240 GOTO 400
300 INPUT "INTRODUZCA LADO Y ANGULO: ";B,AB
310 LET C=B*TAN(AB)
320 LET A=SQR(B^2+C^2)
```

```
330 LET AA=ATN(B/C)
400 PRINT "LOS LADOS SON: ";A;" ";B;" ";C
410 PRINT "LOS ANGULOS SON: ";AA;" ";AB
999 END
```

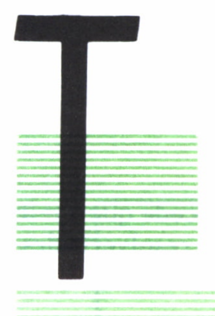
No hemos tenido inconveniente en abusar de IF y GOTO, para simplificar el razonamiento. Bifurca por un MENU y por un ON GOSUB o un ON GOTO. Acostúmbrese a razonamientos simples para mejorarlos con las herramientas que su equipo le permita. Tenga en cuenta que los IF y GOTO existen en todos.

Como en el tema 15 (Espacio vectorial CF) veremos estos conceptos, ahora investigaremos sólo las alturas para el *cálculo de las áreas*. Añadiendo al programa anterior estas instrucciones, obtendremos el valor del área. Ahora bien, como el que trabaja es nuestro micro, aprovecharemos para que nos compare los tres posibles del área; según se calcule con una u otra altura, lógicamente deben coincidir.

```
10 REM CALCULO DEL AREA DE UN TRIANGULO CUALQUIERA
20 INPUT "INTRODUZCA LOS TRES LADOS: ";A,B,C
30 LET P=(A+B+C)/2
40 LET D=P*(P-A)*(P-B)*(P-C)
50 LET S=SQR(D)
60 LET AA=2*S/A
70 LET AB=2*S/B
80 LET AC=2*S/C
100 PRINT "EL AREA CALCULADA CON EL LADO A ES: ";A
A*A
110 PRINT "EL AREA CALCULADA CON EL LADO B ES: ";A
B*B
120 PRINT "EL AREA CALCULADA CON EL LADO C ES: ";A
C*C
```

Si introduce cualquier valor, comprobará cómo coinciden las áreas; en caso contrario, es que hay algún error: verifique sus procesos.

RESOLUCION DE ECUACIONES (I) 8



RATAREMOS de conseguir que nuestro ordenador sea capaz de resolver *ecuaciones de primer grado de una y dos incógnitas.* Lo iniciaremos con algo fácil: sacar una tabla de valores según su ecuación.

```
10 REM TABLA DE VALORES DE UNA FUNCION
20 DEF FNA(X)=2*X-4
30 INPUT "INTRODUZCA HASTA QUE X DESEA: ";N
40 PRINT "X","Y"
50 PRINT "-","-"
60 FOR J=1 TO N
70 PRINT J,FNA(J)
80 NEXT J
999 END
```

Si ejecutamos el programa anterior en nuestro microordenador, obtendremos los siguientes resultados:

```
INTRODUZCA HASTA QUE X DESEA:  10
X             Y
-             -
 1            -2
 2             0
```

3	2
4	4
5	6
6	8
7	10
8	12
9	14
10	16

Supongamos que queremos saber para qué valor de X sucede que Y=0; es decir, cuál es la *solución de la ecuación*. Lo más fácil será resolverla despejando la X:

$$X = -\frac{B}{A}$$

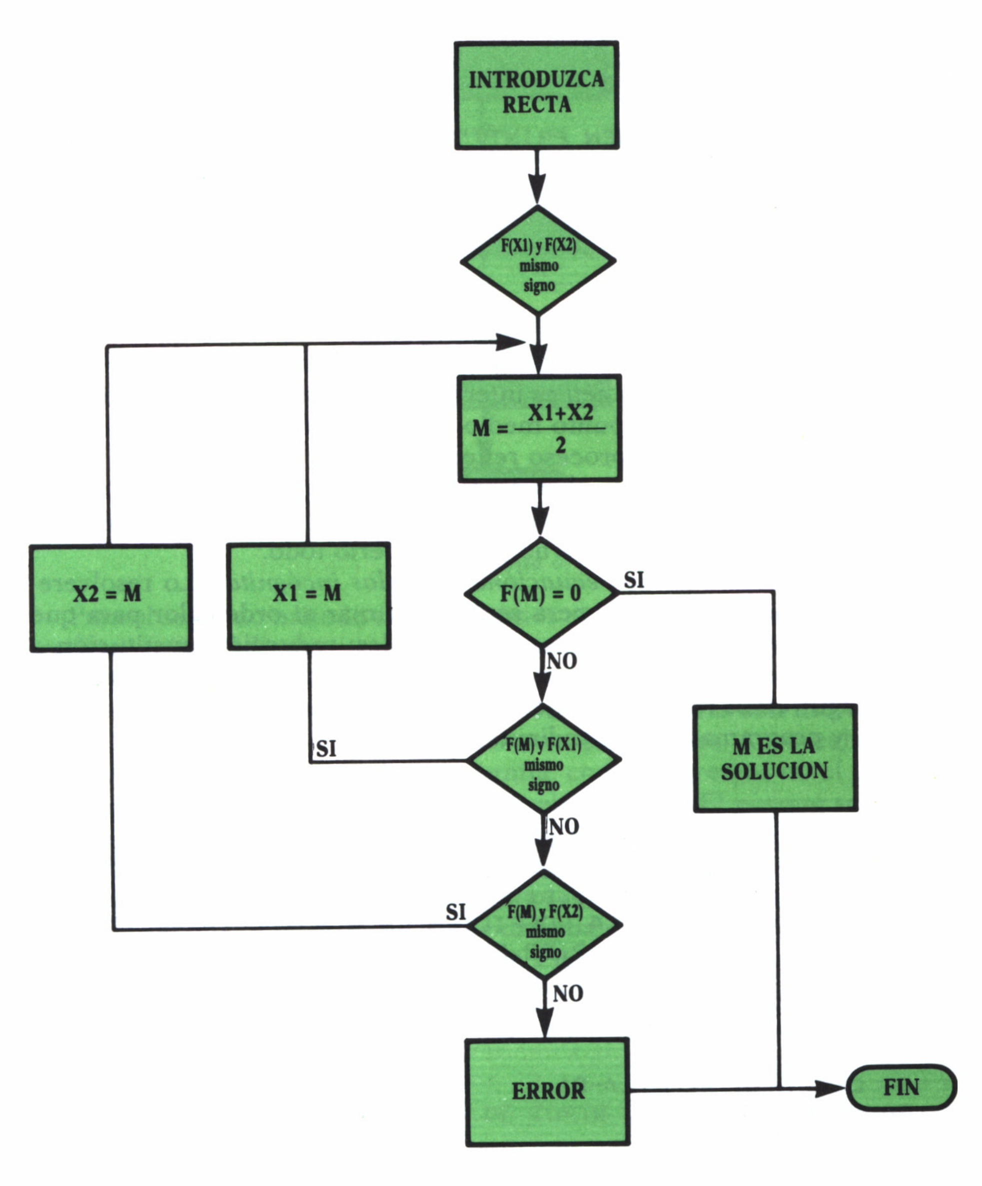
No tendría ningún mérito introducir esta fórmula en nuestro ordenador.

Lo primero, porque hay que buscar una solución válida para otros tipos de ecuaciones, no simplemente para una recta. Y lo segundo, porque no tendría sentido alguno hacer trabajar nuestro ordenador para tal menudencia.

Si el coeficiente de la X no es igual a 0, la recta tiene que cortar en algún punto al eje X. Este punto de corte es la solución de nuestra ecuación. En este punto la recta queda dividida en dos partes: una positiva y otra negativa. ¿Cómo solucionarlo?

Para ello nos basaremos en un método matemático que consiste en tomar dos valores de la recta, uno positivo y otro negativo. Esto querrá decir que la solución se encuentra entre estos dos. Dividimos el intervalo formado por estos dos puntos en dos partes iguales. Calculamos el valor del punto medio; si es negativo, tomaríamos el intervalo formado por el extremo positivo y el punto medio. Si, por el contrario, fuese negativo, el proceso sería inverso. De esta forma, nuestra solución se encontraría en este intervalo, con lo cual hemos reducido a la mitad el número de puntos. Si repetimos este proceso llegará un momento que el intervalo quedará reducido a un punto, ésta es la solución.

Si tuviéramos que hacerlo a mano, nos volveríamos viejos operando. Utilicemos a nuestro ordenador como un trabajador infatigable. El organigrama resultante será:



Organigrama 5

Traspasando el organigrama 5 a programa:

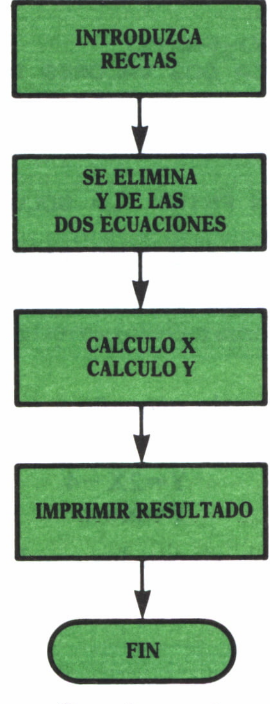
```
10 REM CALCULO DE LA SOLUCION DE LA ECUACION
15 DEF FNA(X)=2*X-4
20 INPUT "INTRODUZCA DOS VALORES DE X: ";X1,X2
30 IF (FNA(X1)*FNA(X2))>0 THEN PRINT "TIENEN EL MI
SMO SIGNO":GOTO 20
40 LET M=(X1+X2)/2
45 PRINT FNA(M)
50 IF FNA(M)=0 THEN PRINT "LA SOLUCION ES: ";M:GOT
O 999
60 IF (FNA(X1)*FNA(M))>0 THEN X1=M:GOTO 40
70 X2=M:GOTO 40
999 END
```

Quizá no vea práctico este programa para una ecuación tan sencilla, pero si lo aplica a una función más complicada podrá obtener los puntos de corte con el eje X. También es interesante la selección de un intervalo u otro, según el valor del punto medio; esto le ayudará a la hora de elección de un camino en un proceso reiterativo.

Notará que preguntamos el intervalo inicial; lógicamente, si estos dos datos iniciales dan valores de Y del mismo signo, no nos sirven; habrá que introducir otros: nuestro micro no puede saberlo todo.

Pasemos al caso de *dos ecuaciones con dos incógnitas.* Lo resolveremos de dos maneras. La primera será programar al ordenador para que sea capaz de utilizar los métodos tradicionales: reducción, sustitución e igualación. Lo prepararemos para reducción. Para crear la lista de instrucciones seguiremos los pasos matemáticos del método. Fíjese en el organigrama 6 y programa correspondiente:

```
10 REM METODO REDUCCION
20 INPUT "INTRODUZCA COEFICIENTES A,B Y C (AX+BY+C
) DE LA 1 RECTA: ";A1,B1,C1
30 INPUT "INTRODUZCA COEFICIENTES A,B Y C (AX+BY+C
) DE LA 2 RECTA: ";A2,B2,C2
40 LET A1=A1*(-B2):LET C1=C1*(-B2)
50 LET A2=A2*(B1):LET C2=C2*B2
60 LET A=A1+A2:LET C=C1+C2
70 LET X=-C/A
80 LET Y=(A1*X+C1)/-B1
90 PRINT "X=";X
100 PRINT "Y=";Y
110 END
```



Organigrama 6

Para el segundo método nos basaremos en el razonamiento hecho para una incógnita. Indicamos que la solución del sistema consistía en dividir la recta en dos partes de distinto signo. Aquí la comparación es entre dos rectas. Antes del punto de corte, los valores de una recta son mayores que la otra; a partir de la solución, se invierten los papeles. Aprovechando la deducción del método para una incógnita, comprobaremos no el valor de Y, sino el valor de la diferencia de los valores de *Y*. El porqué se debe a lo dicho anteriormente: si restamos los valores de Y antes del corte, las diferencias serán todas negativas o positivas, y a partir de éste, tomarán los valores contrarios; luego el razonamiento es válido; tendremos que ir partiendo los intervalos hasta conseguir dar con la solución. Para ver mejor el proceso añadiremos una instrucción PRINT que nos vaya informando de los valores con los que va trabajando el ordenador.

Nuestro programa definitivo:

```
10 REM CALCULO DEL PUNTO DE CORTE DE DOS RECTAS
15 DEF FNA(X)=2*X-4
16 DEF FNB(X)=X+1
```

```
17 DEF FNC(X)=FNB(X)-FNA(X)
20 INPUT "INTRODUZCA DOS VALORES DE X: ";X1,X2
30 IF (FNC(X1)*FNC(X2))>0 THEN PRINT "TIENEN EL MI
SMO SIGNO":GOTO 20
40 LET M=(X1+X2)/2
45 PRINT FNC(M)
50 IF FNC(M)=0 THEN PRINT "LA SOLUCION ES:";M:GOTO
 999
60 IF (FNC(X1)*FNC(M))>0 THEN X1=M:GOTO 40
70 X2=M:GOTO 40
999 END
```

Probémosle con un intervalo pequeño y nos devolvería los siguientes datos para dos rectas fáciles:

$$Y=2X-4$$
$$Y=X+1$$

En el siguiente programa vienen los resultados que nos devuelve el ordenador:

```
INTRODUZCA DOS VALORES DE X:  1.5  3.5
TIENEN EL MISMO SIGNO

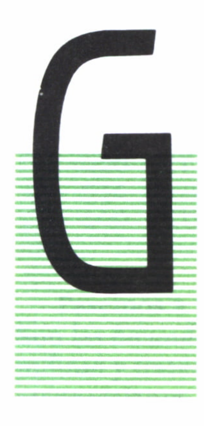
INTRODUZCA DOS VALORES DE X:  5  0
2.5                     2.3841857E-06
1.25                    1.1920928E-06
.625                    5.960464E-07
.3125                   2.980232E-07
.15625                  1.490116E-07
.078125                 7.45058E-08
.0390625                3.72529E-08
.01953125               1.86264E-08
9.765625E-03            9.3132E-09
4.8828125E-03           4.6566E-09
2.44140625E-03          2.3283E-09
1.220703125E-03         1.1641E-09
6.103515625E-04         5.82E-10
3.051757812E-04         2.91E-10
1.525878906E-04         1.455E-10
7.62939453E-05          7.27E-11
3.81469726E-05          3.63E-11
1.90734863E-05          1.81E-11
9.5367431E-06           9E-12
4.7683715E-06           4.5E-12
```

```
2.2E-12                   1E-13
1.1E-12                    0
5E-13                     LA SOLUCION ES: 5
2E-13
```

Esperemos que con estos procesos descritos haya quedado claro cómo conseguir resolver ecuaciones de primer grado.

En sí lo importante no es el resultado, sino el desarrollo empleado para conseguirlo. En temas posteriores resolveremos un sistema de ecuaciones con N incógnitas.

RESOLUCION DE ECUACIONES (II) 9



RACIAS a nosotros, nuestro ordenador va mejorando y ya sabe resolver ecuaciones de primer grado. Instruyámosle más para que sea capaz de resolver ecuaciones de segundo grado.

Para la resolución de estas ecuaciones no vamos a seguir el mismo método que para las de primer grado. Nos fundamentaremos en la fórmula general de resolución de *ecuaciones de segundo grado*. Esta es:

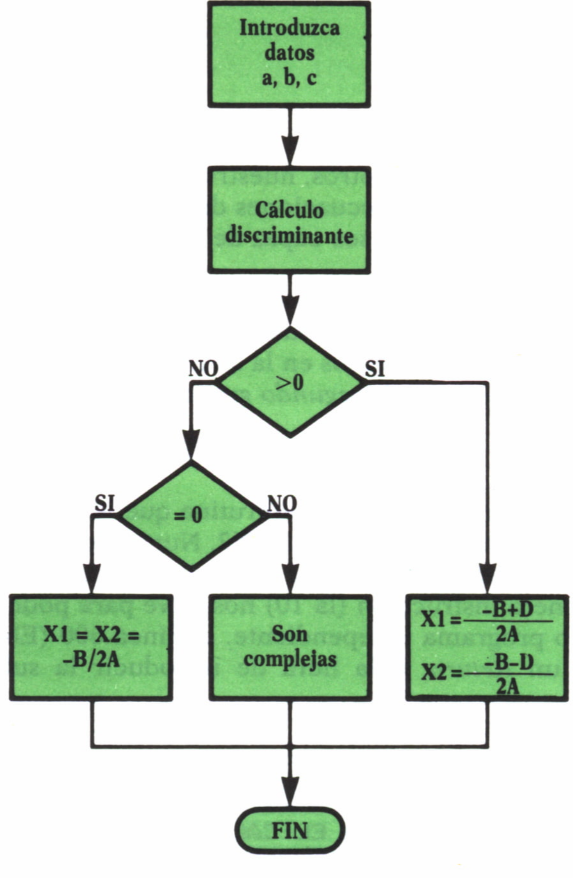
$$X = -B \pm \sqrt{B^2 - 4AC}/2A$$

Empezaremos por preparar una subrutina que, dados los valores a, b, c, nos devuelva las dos soluciones X1 y X2. Numeramos las instrucciones a partir de la 500, para poder llamar a esta subrutina en posteriores programas. La primera instrucción (la 10) nos sirve para poder emplear esta subrutina como programa independiente. La línea 600 (END) deberá ser sustituida por un *Return,* a la hora de introducir la subrutina en un programa:

```
10 INPUT "A,B,C";A,B,C
500 REM SUBRUTINA PARA EL CALCULO DE LAS SOLUCIONES
S DE UNA ECUACION DE SEGUNDO GRADO
510 LET X1=(-B+SQR(B^2-4*A*C))/2*C
520 LET X2=(-B-SQR(B^2-4*A*C))/2*C
530 PRINT "LAS SOLUCIONES SON: ";X1;X2
600 END
```

Como ve, no tiene dificultad alguna. Pruébela con distintos datos y compruebe las soluciones.

Al calcular el *discriminante* (b^2-4ac) puede ocurrir que sea menor que 0. Si se deja así, al ser negativo, nuestro ordenador se quejaría en la función SQR (línea 70), dando error e interrumpiendo el proceso. Es demasiado torpe para mandar un mensaje y continuar la ejecución. Necesita una ayuda por nuestra parte. Como siempre, el hombre es superior a la máquina. Tendríamos que retocar la estructura del programa. Analicémoslo con el organigrama 7, donde se examinan todos los posibles casos.



Organigrama 7

El planteamiento estudiado en el organigrama es el correcto; están tomadas en cuenta todas las posibilidades (incluida la solución compleja). En este último caso se producirían dos soluciones conjugadas, que vendrán dadas en las líneas 510 y 520. En los otros dos casos utilizamos las fórmulas expresadas en el organigrama.

```
5 REM RESOLUCION DE UNA ECUACION DE SEGUNDO GRADO
10 INPUT "INTRODUZCA A,B,C";A,B,C
20 LET D=B^2-4*A*C
40 IF D>0 THEN GOSUB 500:GOTO 100
50 IF D=0 THEN LET X1=-B/2*A:LET X2=X1:PRINT "LAS
SOLUCIONES SON: ";X1,X2:GOTO 100
60 PRINT  "LAS SOLUCIONES SON COMPLEJAS:"
70 PRINT "X1=";-B/(2*A);"+";SQR(ABS(B^2-4*A*C))/(2
*A);"i"
80 PRINT "X1=";-B/(2*A);"-";SQR(ABS(B^2-4*A*C))/(2
*A);"i"
100 END
500 REM SUBRUTINA PARA EL CALCULO DE LAS SOLUCIONE
S DE UNA ECUACION DE SEGUNDO GRADO
510 LET X1=(-B+SQR(B^2-4*A*C))/2*C
520 LET X2=(-B-SQR(B^2-4*A*C))/2*C
530 PRINT "LAS SOLUCIONES SON: ";X1;X2
600 RETURN
```

Pasemos a ver un caso particular de las ecuaciones de segundo grado. Son las *bicuadradas*. Estas, realmente, son de cuarto grado, pero por su similitud y tratamiento se solucionan basándose en las de segundo grado. El razonamiento es fácil: manejarlas como si fueran ecuaciones de segundo grado y, posteriormente, hallar la raíz cuadrada de cada raíz.

No existe dificultad alguna, exceptuando el caso en el cual las raíces sean complejas. Observará que el programa emite dos soluciones y un mensaje (70 a 90); estas soluciones son las calculadas al aplicar el proceso de una ecuación de segundo grado; posteriormente, habría que aplicar la radicación de complejos (capítulo 5), obteniendo las verdaderas soluciones de la bicuadrada.

```
5 REM RESOLUCION DE UNA ECUACION BICUADRADA
10 INPUT "INTRODUZCA A,B,C";A,B,C
20 LET D=B^2-4*A*C
40 IF D>0 THEN GOSUB 500:GOTO 100
60 PRINT "LAS SOLUCIONES SON COMPLEJAS:"
70 PRINT "X1=";-B/(2*A);"+";SQR(ABS(B^2-4*A*C))/(2
*A);"i"
80 PRINT "X2=";-B/(2*A);"-";SQR(ABS(B^2-4*A*C))/(2
*A);"i"
90 PRINT "APLICAR LA RADICACION DE COMPLEJOS PARA
HALLAR"
95 PRINT "LAS VERDADERAS SOLUCIONES"
96 GOTO 200
```

```
100 PRINT "LAS CUATRO SOLUCIONES SON:"
110 IF X1>0 THEN PRINT "X1=";SQR(X1):PRINT "X2=";-
SQR(X1)
120 IF X2>0 THEN PRINT "X3=";SQR(X2):PRINT "X4=";-
SQR(X2)
130 IF X1<0 THEN PRINT "X1=";SQR(ABS(X1));"i"
140 IF X1<0 THEN PRINT "X2=";-SQR(ABS(X1));"i"
150 IF X2<0 THEN PRINT "X3=";-SQR(ABS(X2));"i"
160 IF X2<0 THEN PRINT "X4=";SQR(ABS(X2));"i"
200 END
500 REM SUBRUTINA PARA EL CALCULO DE LAS SOLUCIONE
S DE UNA ECUACION DE SEGUNDO GRADO
510 LET X1=(-B+SQR(B^2-4*A*C))/2*C
520 LET X2=(-B-SQR(B^2-4*A*C))/2*C
530 PRINT "LAS SOLUCIONES SON: ";X1;X2
590 IF D=0 THEN LET X1=-B/2*A:LET X2=X1:PRINT "LAS
 SOLUCIONES SON: ";X1,X2:GOTO 100
600 RETURN
```

Esperemos que no haya tenido ninguna duda.

Existe otro tipo de ecuaciones: las trigonométricas. Son aquéllas en que la incógnita no es X, sino una función trigonométrica de X. El proceso es similar en las bicuadradas. Se trata la función como si fuera de segundo grado. Con las soluciones se calcula el arco que corresponda. Por ejemplo (si es Tg(X)), sería ATM(X1) y ATN(X2).

¡Intente hacerlo solo!

MATRICES 10



N el tratamiento de las matrices está especialmente indicado que intervenga el ordenador. La explicación es muy lógica; los procesos son simples, pero enormemente tediosos y en este tipo de procesos es donde se obtienen las mayores ventajas de un ordenador, por su rapidez de cálculo. Los procesos matriciales se centran siempre en un elemento determinado; posteriormente se aplica a cada uno de los distintos elementos de la matriz. Por tanto, todo se reduce a programar al ordenador, resolviendo el problema concreto de uno de los componentes de la matriz para después, por medio de los FOR'S, extenderlo a los demás.

Aplicando lo expuesto, calcularemos la *transpuesta* de una matriz. Simplemente se trata de intercambiar las filas con las columnas. Siguiendo nuestro razonamiento inicial, centremos la atención en un elemento determinado:

```
10 REM CALCULO DE LA TRANSPUESTA
20 INPUT "INTRODUZCA DIMENSION DE LA MATRIZ ";M,N
30 DIM A(M,N):DIM T(M,N)
40 PRINT "ELEMENTOS DE LA MATRIZ:"
50 FOR J=1 TO N
60 PRINT "FILA ";J
70 FOR K=1 TO M
80 INPUT A(J,K)
90 LET T(K,J)=A(J,K)
100 NEXT K
110 NEXT J
120 PRINT "LA TRANSPUESTA ES: "
130 FOR J=1 TO M
140 PRINT "FILA ";J
```

```
150 FOR K=1 TO N
160 PRINT T(J,K)
170 NEXT K
180 NEXT J
999 END
```

En este programa se aprecian las tres fases: recogida de datos (40-90), aplicación a un elemento (80) e impresión (120 a 170). No tiene ninguna dificultad, ya que sólo hay que igualar el elemento de la matriz a otro, cuyos índices estén intercambiados.

Continuemos operando con estas matrices. La *suma y diferencia de dos matrices* se obtiene sumando, o restando, un elemento con su correspondiente en la otra. Lógicamente deben tener las mismas dimensiones. El programa resultante quedaría:

```
10 REM CALCULO DE LA SUMA
20 INPUT "INTRODUZCA LA DIMENSION DE LAS MATRICES:
 ";N,M
30 DIM A(N,M):DIM B(N,M)
40 PRINT "ELEMENTOS DE LA MATRIZ NUMERO 1:"
50 FOR J=1 TO N
60 PRINT "FILA ";J
70 FOR K=1 TO M
80 INPUT A(J,K)
100 NEXT K
110 NEXT J
120 PRINT "ELEMENTOS DE LA MATRIZ NUMERO 2: "
130 FOR J=1 TO N
140 PRINT "FILA ";J
150 FOR K=1 TO M
160 INPUT B(J,K)
170 NEXT K
180 NEXT J
190 PRINT "LA SUMA ES:"
200 FOR J=1 TO N
210 PRINT "FILA ";J
220 FOR K=1 TO M
230 PRINT A(J,K)+B(J,K)
240 NEXT K
250 NEXT J
```

Queda para el lector insertar instrucciones que controlen posibles errores en los datos, tanto de dimensión como de elementos.

Como prueba de lo indicado al principio del tema, sólo hay que estudiar el problema para un elemento (instrucción 230), aplicando los FOR'S para operar con todos los términos.

La *multiplicación* de matrices constituye un caso interesante del manejo de tablas. Aquí cada elemento resultante es suma de los productos de los términos de la fila I con los de la columna I de la otra matriz. Pasando al programa:

```
10 REM MULTIPLICACION DE MATRICES
20 INPUT "INTRODUZCA DIMENSION DE MATRIZ 1: ";F1,C 1
30 INPUT "INTRODUZCA DIMENSION DE MATRIZ 2: ";F2,C 2
40 IF C1<>F2 THEN PRINT "NO ES POSIBLE LA MULTIPLI
CACION":GOTO 999
45 DIM A(F1,C1):DIM B(F2,C2):DIM P(F1,C2)
50 PRINT "ELEMENTOS DE LA MATRIZ NUMERO 1:"
60 FOR J=1 TO F1
70 PRINT "FILA ";J
80 FOR K=1 TO C1
90 INPUT A(J,K)
100 NEXT K
110 NEXT J
120 PRINT "ELEMENTOS DE LA MATRIZ NUMERO 2:"
130 FOR J=1 TO F2
140 PRINT "FILA ";J
150 FOR K=1 TO C2
160 INPUT B(J,K)
170 NEXT K
180 NEXT J
190 FOR J=1 TO F1
200 FOR K=1 TO C2
210 FOR L=1 TO C1
220 LET P(J,K)=P(J,K)+A(J,L)*B(L,K)
230 NEXT L
240 NEXT K
250 NEXT J
260 PRINT "LA SOLUCION ES:"
270 FOR J=1 TO F1
280 PRINT "FILA ";J
290 FOR K=1 TO C2
300 PRINT P(J,K)
310 NEXT K
320 NEXT J
999 END
```

Lo más interesante de este programa es la anidación de bucles; es decir, el incluir un bucle dentro de otro y en su interior otro, etc. (190-250). Así se controlan las filas y columnas de las matrices pudiendo aplicar la fórmula de la multiplicación en la 220.

En este programa es necesario el control de la instrucción 40, debido a la condición imprescindible de C1=F2. Sin este requisito el tercer FOR (línea 160) no controlaría todos los elementos, originando errores no deseados.

Para el cálculo de la *matriz inversa* es necesario saber el valor del *determinante*. Se plantea el programa basándose en las propiedades de los determinantes. Si tenemos una matriz A de orden N tal que:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Debemos aprovechar las herramientas matemáticas para lograr triangularizar esta matriz de forma que quede:

$$A' = \begin{pmatrix} a'_{11} & a'_{12} & \dots & a'_{in} \\ 0 & a'_{12} & & \\ 0 & 0 & & \\ 0 & 0 & \dots & a'_{nn} \end{pmatrix}$$

¿Cómo conseguir que nuestro ordenador lo haga?

Centrémonos en un elemento perteneciente al triángulo inferior de la matriz. Este elemento debe convertirse en 0; para ello aplicaremos la siguiente propiedad:

Si una matriz B se forma sumando a una fila cualquiera de A otra fila de A multiplicada por un número real:

$$|B| = |A|$$

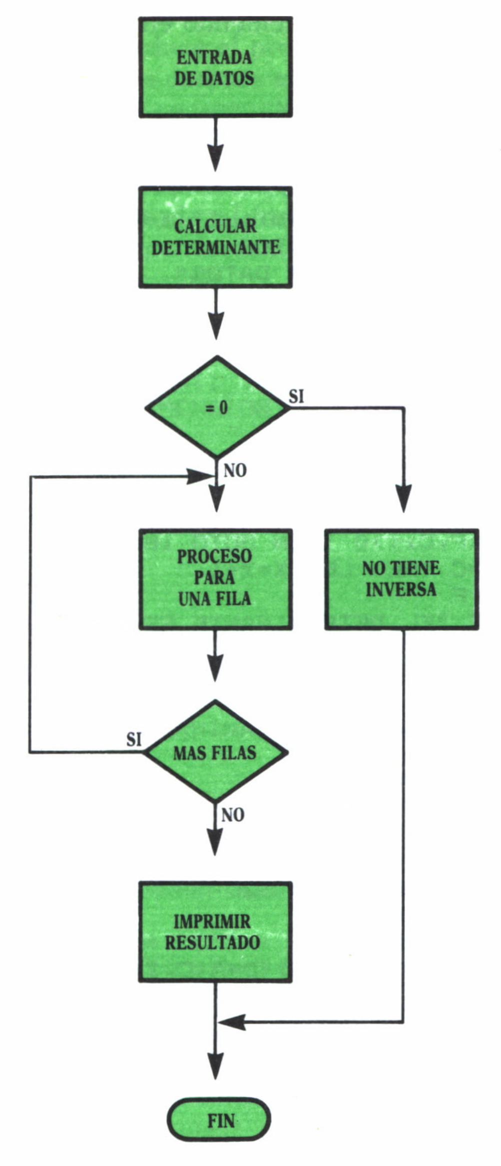
Sea Aij (i>j) perteneciente, lógicamente, a la fila i-columna j. Si a Aij le restamos

$$A_{jj} \cdot \frac{A_{ij}}{A_{jj}}$$

se nos anula. Pero esto mismo, según la propiedad expuesta, ha de aplicarse a toda la fila i. Reiterando el proceso a toda una columna se nos anularán todos los elementos Aij (i>j). Planteando el organigrama 8:

Observe cómo se controla si el determinante es 0. Este caso se producirá cuando los elementos de una columna, a partir de la diagonal incluida, sean nulos.

También hay que tener en cuenta el posible inconveniente de que sea nulo un elemento de la diagonal; en este caso se aplica otra propiedad.



Organigrama 8

Si una matriz B se forma a partir de una matriz A mediante el intercambio de las filas cualesquiera:

$$|B| = -|A|$$

El programa resultante, teniendo en cuenta todas estas consideraciones, es el siguiente:

```
10 REM DETERMINANTE DE UNA MATRIZ CUADRADA
20 INPUT "INTRODUCIR DIMENSION DE LA MATRIZ CUADRA!A
DA A: ";N
30 DIM A(N,N):DIM S(N):LET X=0
40 REM ENTRADA DE DATOS
50 PRINT "INTRODUCIR DATOS"
60 FOR F=1 TO N
70 PRINT "FILA ";F
80 FOR C=1 TO N
90 INPUT A(F,C)
100 NEXT C
105 NEXT F
110 REM CALCULO DEL DETERMINANTE
120 FOR C=1 TO N
130 FOR F=C TO N
140 IF A(F,C)<>0 THEN GOTO 180
150 IF F=C THEN LET X=X+1
160 NEXT F
170 PRINT "EL DETERMINANTE ES NULO":GOTO 999
180 FOR J=1 TO N
190 LET CA=A(C,J):LET A(C,J)=A(F,J):LET A(F,J)=CA
200 NEXT J
210 LET S(C)=A(C,C)
220 LET D=1/A(C,C)
230 FOR K=1 TO N
240 LET A(C,K)=D*A(C,K)
250 NEXT K
260 FOR F=1 TO N
270 IF F=C THEN GOTO 320
280 LET D=-A(F,C)
290 FOR I=1 TO N
300 LET A(F,I)=A(F,I)+D*A(C,I)
310 NEXT I
320 NEXT F
330 NEXT C
340 LET P=1
350 FOR J=1 TO N
360 LET P=P*S(J)
370 NEXT J
380 PRINT "EL VALOR DEL DETERMINANTE ES: ";P*(-1)^X
999 END
```

Se observan varias partes:

1.ª De la 60 a la 100 se introducen los datos; este proceso es similar a los demás programas del capítulo.

2.ª De la 130 a la 170 se comprueba si es nulo el determinante.

3.ª A partir de la 180 se produce el proceso de triangulación. En la 220 se halla el valor de S(C), para después aplicarlo a cada fila utilizando como coeficiente a multiplicar el conseguido en la 240. Este se extenderá a los demás elementos de la fila en el FOR comprendido desde 290 a la 320.

Para el cálculo de la inversa el proceso es idéntico; simplemente añadiremos unas líneas de programa, de tal forma que los procesos efectuados a la matriz se reflejen en la matriz identidad. Al final, la matriz resultante de estas operaciones es la inversa.

```
10 REM CALCULO DE LA MATRIZ INVERSA
20 INPUT "INTRODUCIR DIMENSION DE LA MATRIZ CUADRA
DA A: ";N
30 DIM A(N,N):DIM I(N,N):LET X=0
40 REM ENTRADA DE DATOS
50 PRINT "INTRODUCIR DATOS"
60 FOR F=1 TO N
70 PRINT "FILA ";F
80 FOR C=1 TO N
90 INPUT A(F,C)
95 LET I(F,F)=1
100 NEXT C
105 NEXT F
110 REM CALCULO DEL DETERMINANTE
120 FOR C=1 TO N
130 FOR F=C TO N
140 IF A(F,C)<>0 THEN GOTO 180
150 IF F=C THEN LET X=X+1
160 NEXT F
170 PRINT "NO TIENE INVERSA":GOTO 999
180 FOR J=1 TO N
190 LET CA=A(C,J):LET A(C,J)=A(F,J):LET A(F,J)=CA
195 LET C1=I(C,J):LET I(C,J)=I(F,J):LET I(F,J)=C1
200 NEXT J
210 LET S(C)=A(C,C)
220 LET D=1/A(C,C)
230 FOR K=1 TO N
240 LET A(C,K)=D*A(C,K)
245 LET I(C,K)=D*I(C,K)
250 NEXT K
260 FOR F=1 TO N
270 IF F=C THEN GOTO 320
280 LET D=-A(F,C)
290 FOR I=1 TO N
300 LET A(F,I)=A(F,I)+D*A(C,I)
305 LET I(F,I)=I(F,I)+D*I(C,I)
```

```
310 NEXT I
320 NEXT F
330 NEXT C
340 PRINT "LA INVERSA ES:"
350 FOR F=1 TO N
360 PRINT "FILA ";F
370 FOR C=1 TO N
380 PRINT I(F,C)
390 NEXT C
400 NEXT F
999 END
```

SISTEMAS DE ECUACIONES DE N INCOGNITAS 11



RACIAS al desarrollo de los procesos matemáticos matriciales descritos en el capítulo anterior, se puede resolver con relativa simplicidad el problema de los sistemas de N ecuaciones con N incógnitas. Dado un sistema representado por:

$$a_{11}X_1+a_{12}X_2+a_{13}X_3+ \ldots +a_{14}X_n=C_1$$
$$a_{21}X_1+a_{22}X_2+a_{23}X_3+ \ldots +a_{24}X_n=C_2$$
$$a_{31}X_1+a_{32}X_2+a_{33}X_3+ \ldots +a_{34}X_n=C_3$$
$$\ldots \qquad \ldots$$
$$a_{n1}X_1+a_{n2}X_2+a_{n3}X_3+ \ldots +a_{nn}X_n=C_n$$

Conseguir los valores X1, X2, ..., Xn que satisfagan la igualdad es tarea para nuestro ordenador. Para ello, se tratará de construir programas a imagen y semejanza de los métodos matemáticos existentes.

Hay que tener cuidado en controlar con precisión los bucles necesarios para el desarrollo de los métodos. Es importante no confundir los índices de las FOR, sabiendo en todo momento qué clase de variable representa cada uno, ya sea el número de la columna o el de la fila.

Antes de empezar convendría tener en cuenta una serie de propiedades utilizadas en los razonamientos. Estas son:

- Un sistema no varía si se intercambian dos ecuaciones entre sí.
- Se puede multiplicar todos los términos de una ecuación por un número real distinto de 0.
- Estudiaremos dos métodos conocidos: el de Gauss y el de Gauss-Jordan. Empecemos con el segundo de ellos.



METODO DE GAUSS-JORDAN

Este se basa en conseguir un sistema de la forma:

$$
\begin{aligned}
a_{11}X_1+a_{12}X_2+a_{13}X_3+ \dots +a_{1n}X_n&=C_1\\
a_{22}X_2+a_{23}X_3+ \dots +a_{2n}X_n&=C_2\\
\dots\dots\dots\dots &= \dots\\
a_{n-1,\,n-1}X_{n-1}+a_{n-1,\,n}X_{n-1}&=C_{n-1}\\
a_{n,\,n}X_n&=C_n
\end{aligned}
$$

Si se tuviera un sistema de este tipo, la solución sería trivial. Partiendo de la última ecuación se despeja X_n:

$$X_n = \frac{C_n}{A_{n,\,n}}$$

Llevando este valor a la penúltima se calcula X_{n-1}, y repitiendo este proceso de abajo a arriba, se logrará hallar los restantes valores deseados.

Pero el conjunto del que partimos no es este sistema triangular, sino el siguiente:

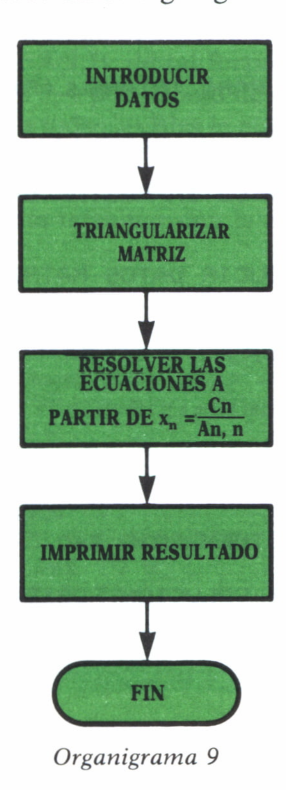
$$
\begin{aligned}
&a_{11}X_1+a_{12}X_2+a_{13}X_3+a_{14}X_4+ \dots +a_{1n}\ X_n=C1\\
&a_{21}X_1+a_{22}X_2+a_{23}X_3+a_{24}X_4+ \dots +a_{2n}\ X_n=C2\\
&\qquad\dots \qquad\qquad\qquad\qquad \dots\\
&a_{n1}X_1+a_{n2}X_m+a_{n3}X_3+a_{n4}X_4+ \dots +a_{nn}\ X_n=CN
\end{aligned}
$$

Eliminando las variables de este sistema debemos construir su matriz correspondiente y, añadiendo la columna de los términos independientes, nos resultaría:

$$
\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \dots & a_{1n}\\
a_{21} & a_{22} & a_{23} & a_{24} & \dots & a_{2n}\\
. & . & . & . & & .\\
a_{n1} & a_{n2} & a_{n3} & a_{n4} & \dots & a_{nn}
\end{pmatrix}
\qquad
\begin{pmatrix}
C1\\ C2\\ .\\ CN
\end{pmatrix}
$$

Si se logra triangularizar esta matriz la tendríamos preparada para obtener la solución del sistema por el método de Gauss-Jordan. Triangularizar una matriz es un problema que vimos en el tema anterior: aquí el método es semejante, sólo se diferencia en pequeños detalles. Lo primero es ver si el sistema es compatible, es decir, si tiene solución. La compatibilidad de un sistema se puede analizar calculando su determinante. Si éste es nulo, el sistema es incompatible. Los elementos que componen el determinante son los coeficientes de las variables a hallar.

Esquematizando el proceso en el organigrama 9.



Organigrama 9

Cada rectángulo representa un proceso diferente; todos ellos encadenados resuelven el problema planteado.

El programa resultante va a ser especial. Se aplicará el método estructurado de Bertini para programarlo. Si no está usted habituado a la programación estructurada no se asuste: simplemente este programa va a ser aparte de una aplicación matemática, un ejemplo de programación estructural. Sin profundizar en el tema, lo esencial del método de Bertini es el plantear el problema subdividiéndolo en módulos. De tal forma que cada parte o subrutina nos resuelve de manera sencilla una parte del programa

principal. Este se compondrá, lógicamente, de varios GOSUB. La lista de instrucciones principal sería:

```
10 REM CALCULO DEL METODO DE GAUSS-JORDAN
20 INPUT "INTRODUZCA VALOR DE N:";N
30 DIM A(N,N+1):DIM S(N)
40 REM SUBRUTINA DE ENTRADA DE DATOS
50 GOSUB 200
60 REM SUBRUTINA DE COMPROBACION SI EL DETERMINANT
E ES CERO
70 GOSUB 300
80 REM SUBRUTINA PARA TRIANGULARIZAR LA MATRIZ
90 GOSUB 400
100 REM SUBRUTINA PARA DESPEJAR LOS VALORES DE X
110 GOSUB 500
120 REM SUBRUTINA DE IMPRESION DE RESULTADOS
130 GOSUB 600
150 END
200 PRINT "INTRODUCIR DATOS MATRIZ DEL SISTEMA"
210 FOR F=1 TO N
220 PRINT "FILA : ";F
230 FOR C=1 TO N+1
240 INPUT A(F,C)
250 NEXT C
260 NEXT F
270 RETURN
300 FOR C=1 TO N
310 FOR F=1 TO N
320 IF A(F,C)<>0 THEN GOTO 360
340 NEXT F
350 PRINT "NO HAY SOLUCION POSIBLE":GOTO 120
360 NEXT C
370 RETURN
```

Los GOSUB 200 y 300 son conocidos por el lector en capítulos anteriores los demás se analizarán uno por uno. Veamos la subrutina de triangularización:

```
400 FOR J=1 TO N
410 FOR C=1 TO N+1
420 LET A(J,C)=A(J,C)/A(J,J)
430 NEXT C
440 FOR L=J+1 TO N
450 FOR C=1 TO N+1
460 LET A(L,C)=A(L,C)-A(L,J)*A(J,C)
```

```
470 NEXT C
480 NEXT L
490 NEXT J
495 RETURN
```

Lo primero es convertir el término A(J,J) en 1. Naturalmente, es dividir por A(J,J), lo cual implica, según las propiedades referidas al principio, que esta operación se repetirá a lo largo de toda la fila.

Una vez conseguida la unidad, se restará a cada fila inferior a J el valor de la fila J multiplicada por A(J,J), de tal forma que todos los elementos pertenecientes a la columna J, y por debajo de la diagonal, serán nulos.

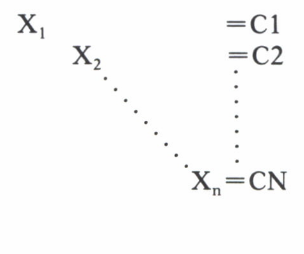
Reiterando este proceso a todas las filas la matriz queda triangularizada.

El proceso de calcular los distintos valores de X es sencillo. Consistirá en un bucle que vaya despejando las ecuaciones en proceso inverso, empezando desde la última hasta terminar con la primera. Las instrucciones serían las siguientes:

```
500 LET S(N)=A(N,N+1)/A(N)
510 FOR J=NT TO 1 STEP -1
520 FOR K=N TO 1 STEP -1
530 LET TO=TO+S(K)*A(N,K)
540 NEXT K
550 LET S(J)=A(J,J+1)/TO
560 NEXT J
570 RETURN
600 PRINT "LA SOLUCION ES:"
610 FOR J=1 TO N
620 PRINT S(J)
630 NEXT J
640 RETURN
```

La rutina de impresión (GOSUB 600) ya es conocida. Es simplemente un FOR que vaya imprimiendo cada uno de los elementos de la matriz X.

Una vez presentado el método de Gauss-Jordan, pasemos al *método de Gauss.* Al igual que en el anterior la clave era triangularizar la matriz, en éste se intenta lograr convertir la matriz de coeficientes en la matriz singular. Quedaría:



Por tanto, una vez realizada esta operación resulta $X_1 = C_1$, $X_2 = C_2$ y así sucesivamente.

Si se comparan los resultados a obtener de uno y otro, se comprueba que los planteamientos serán similares.

Hasta el paso de triangularizar la matriz serán comunes los dos métodos. A partir de aquí empiezan a diferenciarse. Si en el de Gauss-Jordan se despejan las incógnitas, en éste se procederá a triangularizar la matriz superiormente.

De esta forma la matriz quedaría reducida a la diagonal principal, con todos sus elementos igual a 1.

Si se sustituye un proceso por otro, en la operación del ordenador no se advertirán diferencias de tiempo.

El programa del *Método de Jordan* aparece en el siguiente listado:

```
10 REM CALCULO DEL METODO DE GAUSS-JORDAN
20 INPUT "INTRODUZCA VALOR DE N:";N
30 DIM A(N,N+1)
40 REM SUBRUTINA DE ENTRADA DE DATOS
50 GOSUB 200
60 REM SUBRUTINA DE COMPROBACION SI EL DETERMINANT
E ES CERO
70 GOSUB 300
80 REM SUBRUTINA PARA TRIANGULARIZAR LA MATRIZ
90 GOSUB 400
100 REM SUBRUTINA PARA TRIANGULARIZAR LA MATRIZ IN
VERSAMENTE
110 GOSUB 500
120 REM SUBRUTINA DE IMPRESION DE RESULTADOS
130 GOSUB 600
150 END
200 PRINT "INTRODUCIR DATOS MATRIZ SISTEMA"
210 FOR F=1 TO N
220 PRINT "FILA ";F
230 FOR C=1 TO N+1
240 INPUT A(F,C)
250 NEXT C
260 NEXT F
270 RETURN
300 FOR C=1 TO N
310 FOR F=1 TO N
320 IF A(F,C)<>0 THEN GOTO 360
340 NEXT F
350 PRINT "NO HAY SOLUCION POSIBLE":GOTO 120
360 NEXT C
370 RETURN
400 FOR J=1 TO N
410 FOR C=1 TO N+1
```

```
420 LET A(J,C)=A(J,C)/A(J,J)
430 NEXT C
440 FOR L=J+1 TO N
450 FOR C=1 TO N+1
460 LET A(L,C)=A(L,C)-A(L,J)*A(J,C)
470 NEXT C
480 NEXT L
490 NEXT J
495 RETURN
500 FOR J=N TO 1 STEP -1
510 FOR C=N-1 TO 1 STEP -1
520 LET A(J,C)=A(J,C)/A(J,J)
530 NEXT C
540 FOR L=N TO J+1 STEP -1
550 FOR C=N+1 TO 1 STEP -1
560 LET A(L,C)=A(L,C)-A(L,J)*A(J,C)
570 NEXT C
580 NEXT L
590 NEXT J
595 RETURN
600 PRINT "LA SOLUCION ES:"
610 FOR J=1 TO N
620 PRINT A(J,N+1)
630 NEXT J
640 RETURN
```

Como observará, la explicación dada al método de Gauss-Jordan es aplicable a éste. La diferencia viene dada por la línea 500 y sucesivas. En el último método se repite el proceso de triangularización, pero inversamente; esto se consigue intercambiando los límites de los FOR.

Con estos dos métodos se pueden resolver los sistemas de N ecuaciones con N incógnitas. Existe otro llamado de Cramer. Para aplicar este método es necesario:

- Calcular el determinante (A) de la matriz de coeficientes.
- Calcular los determinantes (Ai). Estos (Ai) se obtienen reemplazando en la matriz A la columna i, por el vector de los términos independientes.

Una vez hechos estos cálculos se obtendrían los valores de X_1, X_2, ..., X_n, aplicando las siguientes fórmulas:

$$X_1 = \frac{|A_1|}{|A|}$$

$$X_2 = \frac{|A_2|}{|A|}$$

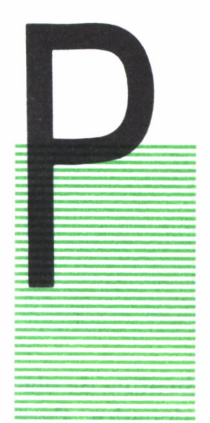
$\Longrightarrow$ en general

$$X_n = \frac{|X_n|}{|A|}$$

Conseguir que el micro sea capaz de hacer uso de este método no representa ninguna dificultad.

Dejo al lector el conseguirlo: tenga en cuenta que todo se basa en hallar los distintos determinantes, simplemente habrá que ir jugando con las columnas respectivas.

COMBINATORIA 12



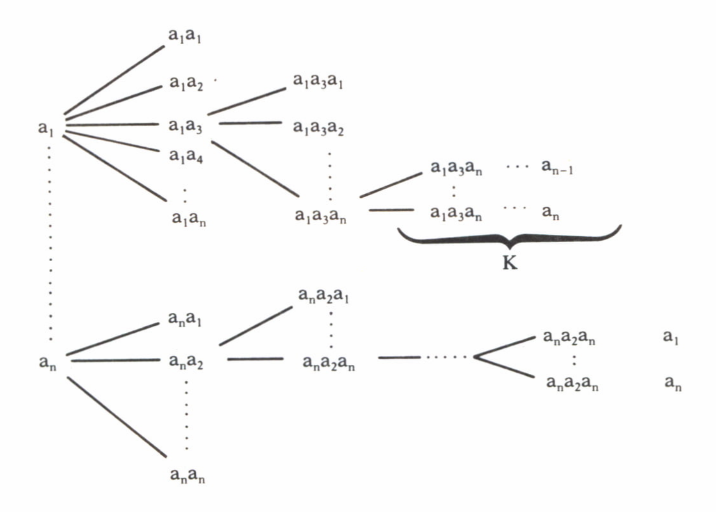
PARA el posterior estudio de estadística y probabilidad, son necesarias las funciones matemáticas tratadas en este capítulo.

El conocimiento del número de casos posibles de un experimento es imprescindible en los cálculos probabilísticos. Empecemos con las *VARIACIONES con REPETICION*. Son necesarios dos datos:

— Número de elementos: (N).
— Cantidad de elementos por combinación: (K).

Sabiendo esto: VRn, $k = n^k$.

A veces es necesario conocer todas las posibles combinaciones. La manera más sencilla es la formación de un árbol. Para ello hacemos:



¿Cómo controlar este árbol con el ordenador?

Para programarlo hay que saber qué orden de variación con repetición se refiere. El programa siguiente controla el caso de N elementos tomados de 4 en 4.

```
10 REM CALCULO DE LAS VARIACIONES CON REPETICION
20 INPUT "NUMERO DE ELEMENTOS:";N
30 DIM A$(N)
40 FOR J=1 TO N
50 INPUT A$(J)
60 NEXT J
70 FOR J=1 TO N
80 FOR K=1 TO N
90 FOR L=1 TO N
100 FOR M=1 TO N
110 PRINT A$(J)+A$(K)+A$(L)+A$(M),
120 NEXT M
130 NEXT L
140 NEXT K
150 NEXT J
999 END
```

Si se introdujeran los datos en orden o se utilizara una subrutina de clasificación al principio, los resultados obtenidos saldrían ordenados. Para modificar este programa, de tal forma que sirva para otro tipo de orden (K), sólo habría que realizar las siguientes rectificaciones:

a) Introducir más FOR, hasta que el número de ellos sea igual a K.
b) En la instrucción 100, añadir tantos A(I) como FOR.

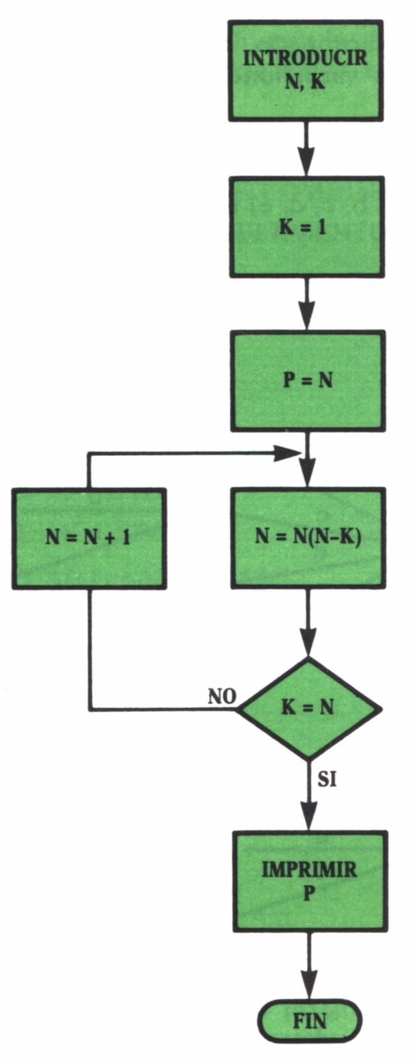
Puede ocurrir que deseemos eliminar las combinaciones en las que se repitan algunos elementos. Estamos hablando de las *variaciones ordinarias*. En ellas, dos variaciones se diferencian en el orden de colocación de los elementos, y en que tienen distintos elementos.

Basándonos en el programa anterior, sólo habría que añadir las siguientes instrucciones:

```
85  IF K=J THEN GOTO 140
95  IF L=K THEN GOTO 130
105 IF M=L THEN GOTO 120
```

Con estas nuevas instrucciones se logra que el ordenador salte al siguiente elemento, cuando reconoce dos iguales. Si en las variaciones con

repetición URn,k= n^k, en las variaciones ordinarias Un,k=N(n−1)(n−2)...(n−(K −1)). Ver organigrama 10.



Organigrama 10

Un caso especial es cuando n=K, estamos entonces en las *permutaciones* de n elementos:

$$Pn=Un,n=n(n-1)(n-2)...3.2.1.$$

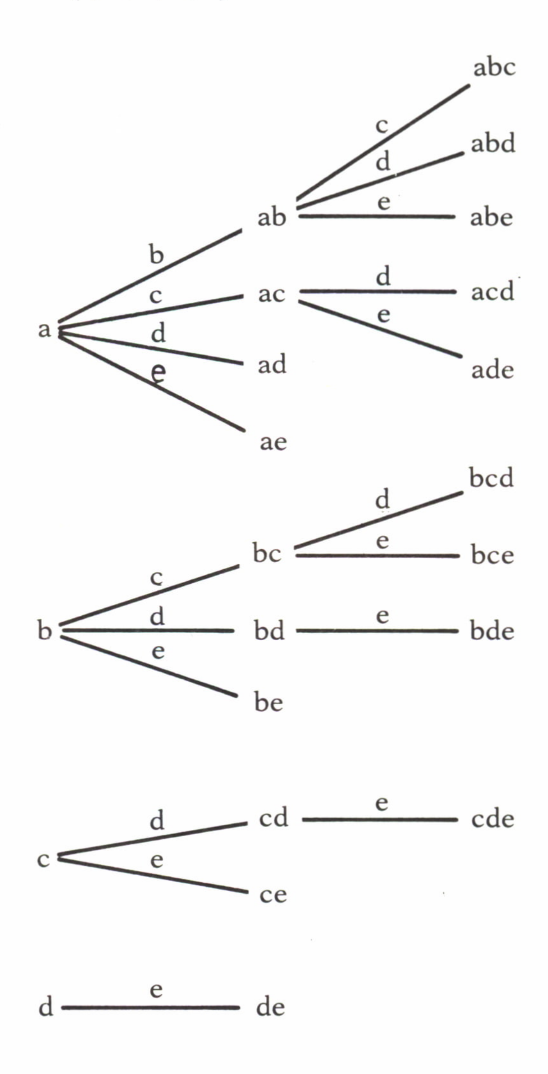
Al producto 1.2.3...(n−2)(n−1).n se denomina factorial de n(n!). Si se quisiera conseguir que el ordenador fuera capaz de calcularlo, simplemen-

te se cambiaría la K por N en el programa anterior. Dejo al lector esta modificación; no obstante, más adelante aplico esta subrutina en diferentes procesos.

Cuando lo que se desea es elegir un número determinado de elementos de un conjunto determinado, estamos hablando de las *COMBINACIONES*. Al igual que las variaciones y permutaciones, existen con y sin repetición.

La formación de las distintas combinaciones daría lugar a un árbol del siguiente tipo:

Supongamos A=(a, b, c, d, e) tomado de 3 en 3.



Como se deduce, una combinación se distingue de otra en los elementos que la integran, *no en el orden de colocación.*

Utilizando el programa aplicado a las variaciones, bastaría rectificar el valor inicial de los índices de los FOR. Cada FOR empezará con el valor de la variable anterior +1, impidiendo de esta forma las posibles repeticiones. El proceso consecuente se refleja en la siguiente lista de instrucciones:

```
10  REM CALCULO DE LAS COMBINACIONES
20  INPUT "NUMERO DE ELEMENTOS";N
30  DIM A$(N)
40  FOR J=1 TO N
50  INPUT A$(N)
60  NEXT J
70  FOR J=1 TO N
80  FOR K=J+1 TO N
90  FOR L=K+1 TO N
100 PRINT A$(J)+A$(K)+A$(L)
110 NEXT L
120 NEXT M
130 NEXT J
```

Si se quisiera saber el número de combinaciones, ésta se halla a partir de la fórmula:

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

Dejo al lector interesado el estudio del programa para este cálculo. Como ayuda indicaré que Cn,k=Cn,n−k, esto es práctico cuando K es grande. Se sustituirá K por n−K, ahorrando cálculos innecesarios.

Si tuviese repeticiones, la fórmula representativa sería:

$$CRN,K=Ck+N-1,K$$

Utilice el cálculo de Cn,K como subrutina y aplíquelo a CRn,K.

Una consecuencia de los números combinatorios es el llamado triángulo de Tartaglia. Este especial triángulo es de la forma:

$$\binom{n}{0}$$

$$\binom{1}{0} \quad \binom{1}{1}$$

$$\binom{2}{0} \quad \binom{2}{1} \quad \binom{2}{2}$$

$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3}$$

....

$$\binom{n}{0} \quad \binom{n}{1} \qquad \qquad \binom{n}{n-1} \quad \binom{n}{n}$$

Sustituyendo éstos por su valor numérico nos quedaría:

$$1$$

$$1 \quad 1$$

$$1 \quad 2 \quad 1$$

$$1 \quad 3 \quad 3 \quad 1$$

$$1 \quad 4 \quad 6 \quad 4 \quad 1$$

....

Lo particular de este triángulo es que cualquier

$$\binom{n}{K} = - \binom{n-1}{K-1} + \binom{n-1}{K}$$

Es decir, cada elemento es la suma de los dos que están sobre él (a derecha e izquierda) en la línea superior. Podremos construir un programa capaz de calcular este triángulo utilizando esta fórmula recursiva. Hay que tener en cuenta los siguientes puntos:

1.º Determinar hasta qué línea queremos obtener el triángulo.

2.º Tener en cuenta que $\binom{n}{o} = 1$ y $\binom{n}{n} = 1$ para cualquier n.

3.º Para poder hacer uso de esta fórmula es necesario ir almacenando los datos anteriores. Para ello guardaremos los resultados en una matriz N×N, para su uso y posterior impresión del triángulo.

```
10 REM CALCULO DEL TRIANGULO DE TARTAGLIA
20 INPUT "INTRODUZCA NUMERO DE FILAS:";N
30 DIM T(N,N+1)
40 FOR J=3 TO N
50 LET T(J,1)=1:LET T(J,N+1)=1
60 LET T(1,1)=1:LET T(2,1)=1:LET T(2,2)=1
70 FOR K=2 TO J
80 LET T(J,K)=T(J-1,K-1)+T(J-1,K)
90 NEXT K
100 NEXT J
110 REM IMPRESION DEL TRIANGULO
120 FOR J=1 TO N
130 FOR K=1 TO J
140 PRINT T(J,K);" ";
150 NEXT K
155 PRINT
160 NEXT J
999 END
```

Fíjese cómo se establecen los límites de los FOR. Debido a que la tabla es N×N, habría filas en las que el recorrido total es innecesario; por ello se limitan hasta K−1. El resultado para un N cualquiera:

```
INTRODUZCA NUMERO DE FILAS: 10
1
1  1
1  2   1
1  3   3   1
1  4   6   4   1
1  5   10  10  5   1
1  6   15  20  15  6   1
1  7   21  35  35  21  7   1
1  8   28  56  70  56  28  8   1
1  9   36  84  126 126 84  36  9  1
```

El binomio de Newton es otra aplicación de los números combinatorios. Este desarrollo utilizado para el cálculo de la potencia M de un binomio viene dado por:

$$(X+A)^n = \binom{n}{0} A^n \cdot X^0 + \binom{n}{1} A^{n-1} \cdot X^1 + \ldots + \binom{n}{n} \cdot X^n$$

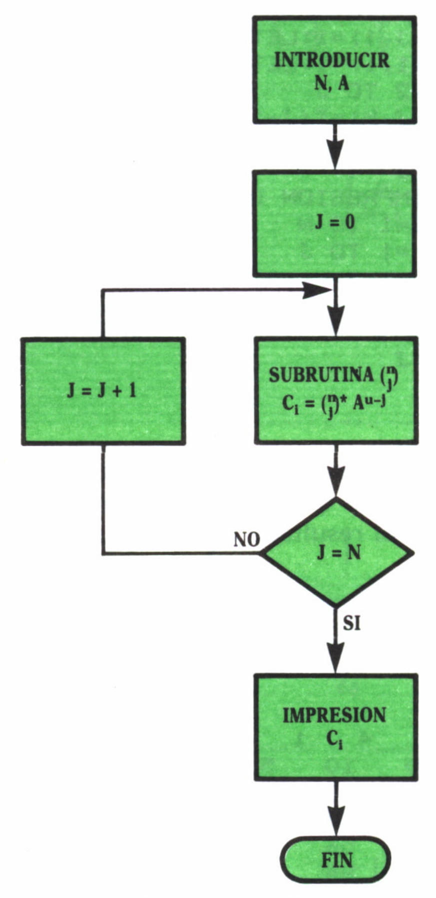
o bien de manera simplificada:

$$(X+A)^n = \Sigma \binom{n}{i} A^{n-i} \cdot X^i$$

A partir de este término general y aplicando lo aprendido, se logrará que el ordenador sepa elevar a una potencia M cualquier binomio. Tendremos en cuenta las siguientes consideraciones:

a) Como aplicación concreta utilizaremos un binomio del tipo (X+A).

b) El cálculo del número combinatorio se hará por medio de una subrutina.



Organigrama 11

El programa deducido del organigrama 11:

```
10 REM CALCULO DEL BINOMIO DE NEWTON
20 INPUT "INTRODUCIR A(X+A): ";A
30 INPUT "INTRODUCIR EXPONENTE: ";N
40 DIM C(N+1)
50 FOR J=1 TO N
60 GOSUB 200
70 LET C(J)=A^(N+1-J)*CO
80 NEXT J
90 REM IMPRESION DE LOS RESULTADOS
100 FOR J=1 TO N+1
110 PRINT "COEFICIENTE DE X^";J-1;" ES ";C(J)
120 NEXT J
130 END
200 REM SUBRUTINA DE FACTORIAL DE UN NUMERO COMBIN
ATORIO
210 LET CO=1
220 LET F1=1
230 FOR L=1 TO J-1
240 LET F1=F1*L
250 NEXT L
260 LET F2=1
270 FOR P=1 TO N+1-J
280 F2=F2*P
290 NEXT P
300 LET FA=1
310 FOR M=1 TO N
320 LET FA=FA*M
330 NEXT M
340 LET CO=FA/(F1*F2)
350 RETURN
```

Dejo al lector un caso interesante: (A+Bi)? Como sugerencia indicaremos que es similar el proceso, sólo habrá que tener cuidado con el signo. (¡Recuerde! que $(i)^{2n}=1$ y $(i)^{2n+1}=-1$.)

ESTADISTICA (I) 13

CUANDO Herman Holletith en 1890 ideó la actualmente superconocida tarjeta perforada, lo hizo con un fin: realizar el censo. Es decir, intentar mecanizar los procesos estadísticos. Tenía sobradas razones para ello: los cálculos estadísticos son sencillos pero numerosos. En el momento que la recogida de datos es amplia, las operaciones se multiplican.

A lo largo de los dos capítulos siguientes se tratará de aprovechar la cualidad más importante de nuestro ordenador (la rapidez al operar) para nuestros cálculos estadísticos. Iniciaremos hallando las medidas centrales y presentando las representaciones gráficas (capítulo 13) y medidas de dispersión (capítulo 14), concluyendo con un análisis estadístico completo.

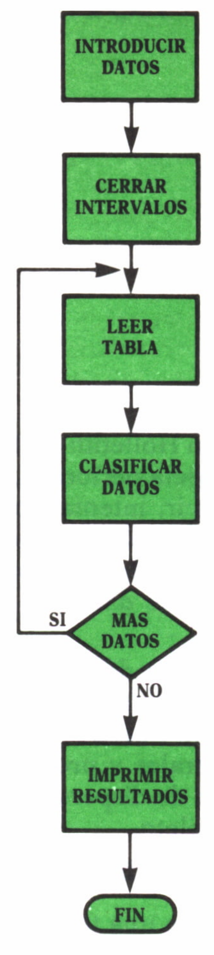
En todos los programas se tendrán las siguientes consideraciones:

- Los datos se almacenarán en una tabla (en realidad se leerían de un fichero). De todas formas, los procesos son similares.
- Todos los cálculos son relativos al espacio muestral. Por tanto, será necesario ir acumulando los distintos valores aplicables a estos cálculos para su posterior uso.

No se controlan los datos de entrada, se darán por revisados. Esto ocurre en la realidad; la verificación de datos suele ser antes de los procesos automatizados.

Comencemos con los cálculos de las frecuencias. Estas no son más que el número de veces que se repite determinada variable.

Para la *frecuencia absoluta* se van incrementando los distintos acumuladores según el dato leído. Como no se puede tener indefinidos acumuladores hace falta limitarlos. Observe el organigrama 12.



Organigrama 12

Estadísticamente se utiliza los intervalos de clase. Si a nuestro ordenador le introducimos los límites inferiores y superiores y, además, la amplitud del intervalo, podremos crear una tabla de acumuladores cuya dimensión

$$N = \frac{\text{Valor máximo-valor mínimo}}{\text{Amplitud del intervalo}}$$

El desarrollo para un dato determinado es claro: comparar el valor del dato con los extremos inferiores y superiores de cada intervalo; cuando se

encuentre el deseado, se incrementa el correspondiente acumulador. El programa resultante:

```
10 REM CALCULO DE LA FRECUENCIA ABSOLUTA
20 INPUT "NUMERO DE DATOS: ";N
30 DIM A(N)
40 PRINT "INTRODUZCA DATOS:"
50 FOR F=1 TO N
60 INPUT A(F)
70 NEXT F
80 INPUT "INTRODUZCA AMPLITUD DE LOS INTERVALOS: "
;A
90 INPUT "INTRODUZCA LOS VALORES MAXIMO Y MINIMO:
";MX,MN
100 LET L=ABS(INT((MX-MN)/A))
110 DIM C(L)
120 FOR J=1 TO N
130 FOR K=1 TO L
140 IF A(J)<(MN+K*A) THEN C(K)=C(K)+1:GOTO 160
150 NEXT K
160 NEXT J
170 REM IMPRESION RESULTADOS
180 PRINT "LA SOLUCION ES:"
186 PRINT "=========","=========="
190 FOR K=1 TO L
200 PRINT MN+(K-1)*A;"-";MN+K*A,C(K)
210 NEXT K
999 END
```

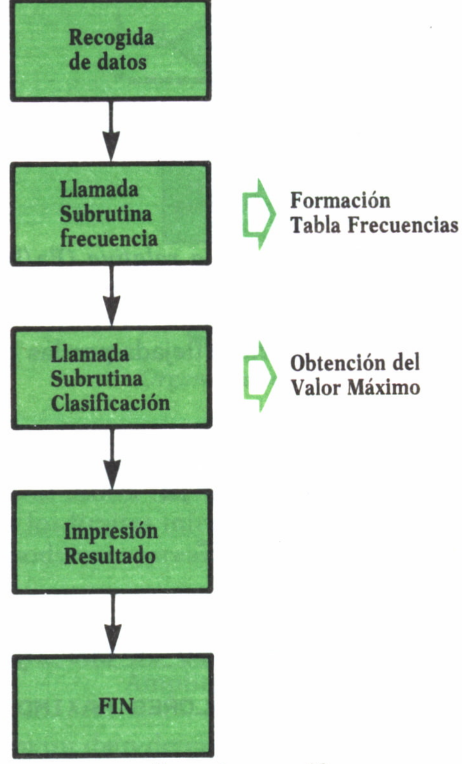
Si se desea el valor de la *frecuencia relativa* (Fa/N), se ampliará este programa, de tal forma que, utilizando los resultados anteriores, obtendríamos los distintos cocientes de las frecuencias relativas.

La ampliación pertinente vendría reflejada por las líneas de instrucciones siguientes:

```
10 REM CALCULO DE LA FRECUENCIA ABSOLUTA
20 INPUT "NUMERO DE DATOS: ";N
30 DIM A(N)
40 PRINT "INTRODUZCA DATOS:"
50 FOR F=1 TO N
60 INPUT A(F)
70 NEXT F
80 INPUT "INTRODUZCA AMPLITUD DE LOS INTERVALOS: "
;A
90 INPUT "INTRODUZCA LOS VALORES MAXIMO Y MINIMO:
";MX,MN
```

```
100 LET L=ABS(INT((MX-MN)/A))
110 DIM C(L)
120 FOR J=1 TO N
130 FOR K=1 TO L
140 IF A(J)<(MN+K*A) THEN C(K)=C(K)+1:GOTO 160
150 NEXT K
160 NEXT J
170 REM IMPRESION RESULTADOS
180 PRINT "LA SOLUCION ES:"
186 PRINT "=========","=========="
190 FOR K=1 TO L
200 PRINT MN+(K-1)*A;"-";MN+K*A,C(K)/N
210 NEXT K
999 END
```

Otra medida interesante son las *frecuencias acumuladas,* tanto relativas como absolutas. El proceso es idéntico, la única variación es la acumulación de las frecuencias anteriores a la hallada. Dejo al lector el cálculo de estas frecuencias.



Organigrama 13

Comencemos con las medidas llamadas centrales. Estas son:

- Media aritmética.
- Moda.
- Mediana.

La *moda m,* de una serie estadística, es el valor de mayor frecuencia.

Para su estudio es necesario dos subrutinas ya vistas: una encargada del cálculo de las frecuencias, vista en este capítulo, y otra de clasificación para obtener la frecuencia máxima (capítulo 1). Estructuraremos nuestro programa de acuerdo al organigrama 13.

El listado resultante y la solución para un caso concreto:

```
 10 REM CALCULO DE LA MODA
 20 INPUT "NUMERO DE DATOS: ";N
 30 DIM D(N)
 40 PRINT "INTRODUZCA DATOS:"
 50 FOR I=1 TO N
 60 INPUT D(I)
 70 NEXT I
 80 INPUT "INTRODUZCA LOS VALORES MAXIMO Y MINIMO:
    ";MX,MN
 90 DIM C(MX+N)
 100 FOR I=1 TO N
 110 IF D(I)>=MN AND D(I)<=MX THEN LET C(D(I))=C(D(I))+1
 120 NEXT I
 130 FOR I=1 TO N+MX
 140 IF C(1)<C(I) THEN LET C(1)=C(I):LET S=I
 150 NEXT I
 160 PRINT "LA MODA ES: ";S
 999 END
```

```
NUMERO DE DATOS:  10
INTRODUZCA DATOS:
?  1
?  2
?  2
?  3
?  5
?  5
?  5
?  5
?  5
?  4
INTRODUZCA A,MPLITUD DE LOS INTERVALOS:  2
```

```
INTRODUZCA LOS VALORES MAXIMO Y MINIMO:  1  6
LA MODA ES: 5.5
```

La *media aritmética* viene dada por la fórmula:

$$MA = \frac{1}{N} \sum_{i=1}^{k} Xi \cdot fi$$

en donde Xi son los distintos valores de X, fi su frecuencia y N el número total de valores, lógicamente N=fi+f2+ ... +fk

Nos encontramos de nuevo con el esquema de trabajo indicado al principio del capítulo. El programa es similar al de la moda, la variación será la sustitución de la subrutina de clasificación por una que calcule el algoritmo de la media. Aprovechando la lista de instrucciones anterior e introduciendo nuestra modificación, el resultado sería:

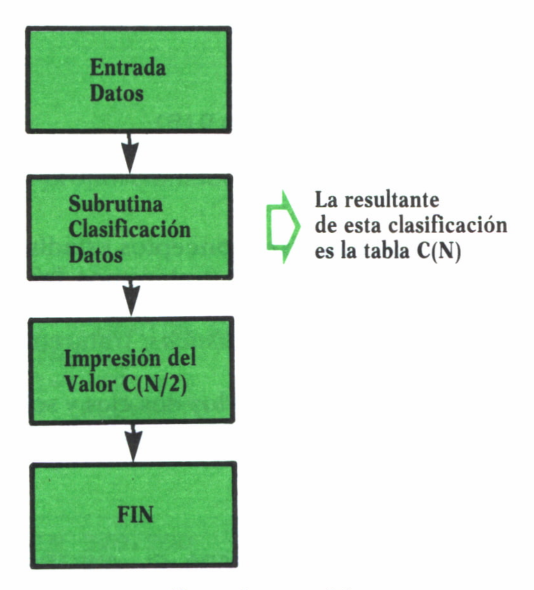
```
10 REM CALCULO DE LA MEDIA ARITMETICA
20 INPUT "NUMERO DE DATOS: ";N
30 DIM D(N)
40 PRINT "INTRODUZCA DATOS:"
50 FOR J=1 TO N
60 INPUT D(J)
70 NEXT J
90 INPUT "INTRODUZCA AMPLITUD DE LOS INTERVALOS: "
;A
100 INPUT "INTRODUZCA LOS VALORES MAXIMO Y MINIMO:
 ";MX,MN
110 LET L=ABS(INT((MX-MN)/A))
115 DIM C(L)
120 FOR J=1 TO N
130 FOR K=1 TO L
140 IF D(J)<(MN+K*A) THEN LET C(K)=C(K)+1:GOTO 160
150 NEXT K
160 NEXT J
170 REM CALCULOS
180 FOR K=1 TO L
190 LET ME=ME+((MN+K*A+A/2)*C(K))
200 NEXT K
210 PRINT "LA MEDIA ES: ";ME/N
999 END
```

Queda por ver la *Mediana*. Su definición estadística es:

Se llama mediana de una serie estadística al valor M, tal que existe igual número de observaciones mayores que menores que M.

Si se observa, no hay más que coger el elemento N/2 de nuestra matriz de datos. Pero si no queremos un valor erróneo, es necesaria una clasificación previa de esta tabla.

El proceso reflejado en el organigrama 14 será más fácil de asimilar:



Organigrama 14

Conocemos, por el capítulo 1, cómo efectuar una clasificación (SORT) de un conjunto determinado. Si se aplica esta subrutina a nuestros datos iniciales, la obtención de la mediana no tiene complicación alguna. Antes de escribir nuestra lista de instrucciones, se debe tomar en consideración un detalle importante. Pese a la rapidez de operación del ordenador, es necesario el ahorro de tiempo, máximo cuando los datos pueden ser numerosos. Se necesita el elemento N/2 de la tabla ya ordenada; si se clasifica hasta dicho elemento ahorraremos la clasificación de los términos posteriores. Así en lugar de N pasadas, bastará N/2 con la consiguiente ganancia de tiempo.

```
10 REM CALCULO DE LA MEDIANA
20 INPUT "NUMERO DE DATOS: ";N
30 DIM D(N)
40 PRINT "INTRODUZCA DATOS:"
50 FOR J=1 TO N
60 INPUT D(J)
```

```
70 NEXT J
80 REM CLASIFICAFION DE LOS DATOS
90 FOR J=1 TO N
100 FOR K=J TO N
110 IF D(J)<D(K) THEN LET X=D(J):LET D(J)=D(K):LET
 D(K)=X
120 NEXT K
130 NEXT J
140 LET S=INT(N/2)+1
150 PRINT "LA MEDIANA ES: ";D(S)
999 END
```

A lo largo del capítulo se han visto conceptos estadísticos de las llamadas medidas centrales. A continuación trataremos las representaciones gráficas más usuales.

Existen varios tipos de *gráficas estadísticas;* trataremos dos de ellas: la de barras y la de sectores.

La de barras consiste en representar los dos ejes y sobre el eje X elevar barras cuya elevación sea el porcentaje de la variable. El programa resultante será:

```
10 REM GRAFICAS DE BARRAS PARA EL AMSTRAD
20 MOVE 0,0
30 INPUT "NUM. DE ELEMENTOS :";N
40 IF N>200 OR N<1 THEN PRINT "ERROR: ENTRE 1 Y 20
0":GOTO 30
50 DIM TABLA(N)
60 INPUT "TECLEE DIFERENCIA ENTRE ESCALAS: ";V
70 FOR I=1 TO N
80 PRINT "DATO ";I:INPUT TABLA(I)
90 IF TABLA(I)>25*V THEN PRINT "ERROR:VALOR SUPERI
OR A";25*V:GOTO 80
100 NEXT I
110 CLS
120 BARRADO=INT(200/N)
140 PLOT 100,200:DRAW 100,400
150 FOR I=BARRADO TO 200 STEP BARRADO
160 PLOT 100+I,195:DRAW 100+I,200
170 NEXT I
180 FOR J=V TO 200 STEP BARRADO
190 PLOT 95,200+J:DRAW 100+I,200
200 NEXT J
210 J=0
220 FOR I=BARRADO TO 200 STEP BARRADO
```

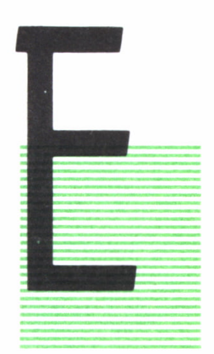
```
230 J=J+1
240 CIMA=25*V
250 PORC=INT(TABLA(J)*100/CIMA)
260 SUBIDA=INT(PORC*CIMA/100)
270 PLOT 100+I,200
280 DRAW 100+I,200+SUBIDA
290 NEXT I
```

Otro tipo de gráficas consiste en dividir la circunferencia en secciones, cuyos ángulos son proporcionales de las variables. El razonamiento es similar al anterior.

```
10 REM GRAFICA DE SECTORES PARA AMSTRAD
20 INPUT "CUANTOS ELEMENTOS: ";N
30 DIM TABLA(N)
40 FOR I=1 TO N
50 PRINT "ELEMENTO ";I:INPUT TABLA(I)
55 NUM=NUM+TABLA(I)
60 NEXT I
70 CLS
80 DEG
90 ORIGIN 320,200
100 FOR I=1 TO 360
110 PLOT 100*COS(I),100*SIN(I)
120 NEXT I
130 DIM GRAD(N)
140 FOR I=1 TO N
150 PORC=INT(TABLA(I)*100/NUM)
160 GRAD(I)=INT(360*PORC/100)
170 NEXT I
200 G=0
210 FOR J=1 TO N
215 ORIGIN 320,200
220 DRAW 100*COS(GRAD(J)+G),100*SIN(GRAD(J)+G)
230 G=G+GRAD(J)
240 NEXT J
```

Existen otros tipos de gráficas; su mayor o menor dificultad dependerán de la pericia del lector. ¡Animo!

ESTADISTICA (II) 14



MEDIDAS DE DISPERSION



EN el capítulo anterior se trató de las medidas centrales y, posteriormente, se logró hacer gráficas estadísticas con el ordenador. Ahora se verán las llamadas *medidas de dispersión*.

La utilidad de éstas viene dada por la necesidad de completar la información conseguida por las medidas centrales. Si se observa la siguiente tabla:

Calificaciones	**fi**
0	**0**
1	**0**
2	**12**
3	**20**
4	**8**
5	**0**
6	**0**

Media $\overline{X}$ = 2,90

Calificaciones	**fi**
0	**4**
1	**11**
2	**5**
3	**4**
4	**3**
5	**9**
6	**4**

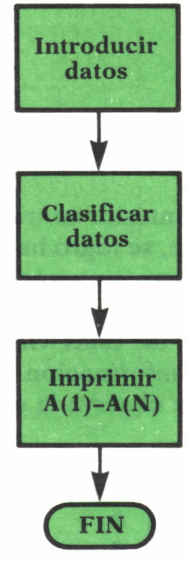
$\overline{Y}$ = 2,85

Se ve cómo las medias son casi iguales, pero en la tabla B, los datos están más dispersos. Por esta razón se pensó en la utilización de otras medidas, que informarán sobre esta dispersión. Las medidas de dispersión son siempre respecto a la media aritmética. Ya tenemos un punto de apoyo a la hora de estructurar los programas. Habrá que echar mano de los procesos y subrutinas del tema anterior para el cálculo de las medidas de dispersión.

La primera en tratar es la *desviación media*. Para ello, vamos a definir el concepto de *Recorrido:*

— Recorrido de una serie estadística es la diferencia entre el valor máximo y el valor mínimo.

Fíjese en el organigrama 15.



Organigrama 15

Tanto la recogida de datos como su posterior clasificación no deben ofrecer dificultad alguna, ya que ha sido tratado en temas anteriores. Espero que el lector no encuentre ningún obstáculo en la confección del correspondiente programa. Lógicamente UMax=A(1) y Umin=A(N), siempre y cuando esté ordenada la tabla de mayor a menor; en caso contrario, la asignación del valor máximo y mínimo sería la inversa.

Una vez conocido el recorrido continuemos con el concepto de *desviación media*.

Al estudiar la media aritmética, se vio que era una medida de tendencia central; pues bien, la desviación media indica el valor en que los diferentes elementos de una serie se alejan de este punto central.

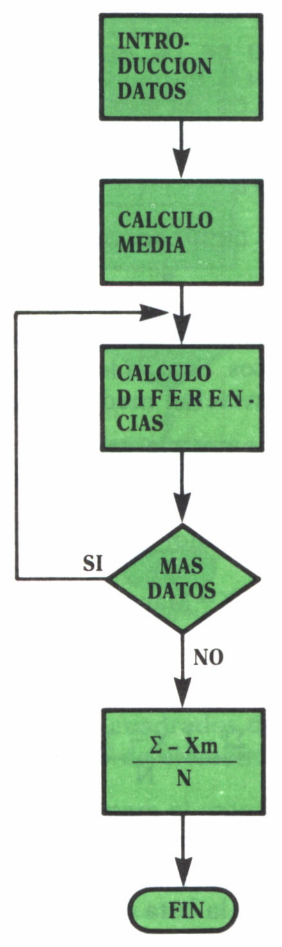
Lógicamente lo ideal es que esta medida fuese igual a 0, pero esto no sucede nunca.

Cuando el espacio muestral es pequeño, la fórmula estadística más apropiada es:

$$DM = \frac{\Sigma \mid X - Xm \mid}{N}$$

en la que:

X = Valor de los distintos elementos.
XM = Valor de la media aritmética.
N = Número de elementos.
Reflejemos el desarrollo en el organigrama 16.



Organigrama 16.

El programa deducido del organigrama 16.

```
10 REM CALCULO DE LA DESVIACION MEDIA
20 REM ENTRADA DE DATOS
```

```
25 LET SU=0:LET DT=0
30 INPUT "NUMERO DE DATOS? ";N
40 DIM A(N)
50 FOR J=1 TO N
60 INPUT A(J)
65 LET SU=SU+A(J)
70 NEXT J
80 REM CALCULO DE LA MEDIA
90 LET ME=SU/N
100 REM CALCULOS DE LAS DESVIACIONES
110 FOR J=1 TO N
120 LET D=ABS(A(J)-ME)
130 LET DT=DT+D
140 NEXT J
150 PRINT "LA DESVIACION MEDIA ES: ";DT/N
```

Cuando el número de datos es numeroso sería conveniente dividir el espacio muestral en intervalos. Esto ya lo hicimos en el tema anterior; ahora bien, el cálculo de la desviación media varía. El camino a seguir sería:

1.º Se distribuye la serie por clases.
2.º Se hallan las frecuencias (X), el número de elementos que pertenecen a cada clase.
3.º Se halla el punto medio, marca de clase, de cada intervalo.
4.º Se halla la media.
5.º Se calcula (X−XM), X=Marca de clase.
6.º Se multiplican las frecuencias por (X−Xm).

La fórmula nos queda:

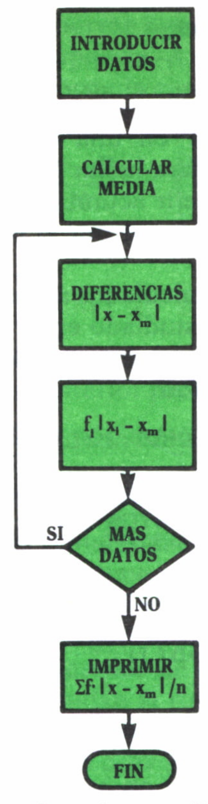
$$\frac{\Sigma f \cdot | X - Xm |}{N}$$

Fíjese en el organigrama 17 e intente construir el programa.

Todos los procesos son conocidos, luego no creo que presente ninguna dificultad la realización de la lista de instrucciones.

Otra medida de dispersión para hallar la homogeneidad más o menos acentuada de un conjunto de elementos es la *desviación típica o estándar*. La fórmula es la siguiente:

$$\sigma = \sqrt{\frac{\Sigma | X - Xm |^2}{N}}$$



Organigrama 17

en la que

$\Sigma \mid X - Xm \mid^2$ = Suma de los cuadrados de las desviaciones.
N = Número de elementos.

Ya que en la desviación media para el caso de pocos elementos, en ésta lo haremos para un gran espacio muestral.

En este caso la fórmula quedaría:

$$\sigma = \sqrt{\frac{\Sigma f \cdot \mid X - Xm \mid^2}{N}}$$

donde:

X = Punto medio de cada intervalo.
F = Frecuencia de cada clase.
N y Xm son el número de elementos y la media aritmética.

Los pasos son los siguientes:

1.º Recogida de datos.
2.º Hallar las frecuencias de cada clase.
3.º Calcular y almacenar en una tabla las marcas de cada clase.
4.º Aplicar la subrutina para hallar la media aritmética.
5.º Hallar $f \cdot |X-Xm|^2$ e ir acumulando estos colores.
6.º Aplicar la fórmula 6 e imprimir el resultado.

Todos los pasos han sido calculados anteriormente, sólo el punto 5.º es diferente, pero es similar al estudiado en el programa anterior.

Fíjese en el programa; los comentarios (REM) irán indicando cada uno de los diferentes pasos a realizar:

```
10 REM CALCULO DE LA DESVIACION TIPICA
20 INPUT "NUMERO DE DATOS";N
25 LET SU=0
30 DIM A(N)
40 PRINT "INTRODUZCA DATOS:"
50 FOR J=1 TO N
60 INPUT D(J):SU=SU+D(J)
70 NEXT J
75 LET ME=SU/N
80 INPUT "INTRODUZCA AMPLITUD DE LOS INTERVALOS:";
A
90 INPUT "INTRODUZCA LOS VALORES MAXIMO Y MINIMO:"
;MX,MN
100 LET L=INT(MX-MN)/A
110 DIM C(L)
120 FOR J=1 TO N
130 FOR K=1 TO L
140 IF D(J)<(MN+K*A) THEN LET C(K)=C(K)+1:GOTO 160
150 NEXT K
160 NEXT J
170 REM FORMACION DE LA TABLA DE LOS PUNTOS MEDIOS
180 DIM M(L)
190 FOR J=1 TO L
200 LET M(J)=MN+A*(J-1)+A/2
210 NEXT J
220 REM CALCULO DEL CUADRADO DE LAS DESVIACIONES
230 FOR J=1 TO L
240 LET D=ABS(M(J)-ME)
250 LET SU=SU+D^2*C(J)
260 NEXT J
270 LET DE=SQR(D/N)
280 PRINT"LA DESVIACION TIPICA ES:"DE
```

Un método estadístico para ahorrar operaciones es aplicar el método abreviado de elegir una media supuesta. A partir de esta media calcular las desviaciones (D) y aplicar la fórmula siguiente:

$$\sigma = I \sqrt{\frac{\Sigma f \cdot d^2}{N} - \left(\frac{\Sigma f \cdot d}{N}\right)^2}$$

en la que

I = Intervalo de clase.
N = Número de elementos.
Simplemente es modificar el anterior programa de la siguiente manera:

1.° Pedir la supuesta media; por tanto, eliminar la subrutina de la media.
2.° Sustituir esta fórmula donde se aplique la otra.

Dejo al lector comprobar la rentabilidad de un método y de otro. Hay que tener en cuenta que siempre es necesario probar los métodos para varios grupos de elementos. Muchas veces el empleo de distintos métodos en procesos informáticos varían según la cantidad de datos a tratar. Un ejemplo claro son los distintos métodos de clasificación; su elección depende de variadas fórmulas que midan la rentabilidad del empleo de uno u otro, en relación al número y tipos de datos.

A veces es interesante conocer la desviación en porcentaje. Habría que añadir a nuestro programa los siguientes puntos:

— Calcular Ma $\pm$ σ.
— Ver cuántos elementos están en (Ma$-\sigma$, Ma$+\sigma$).
— Calcular el porcentaje que representan estos elementos en el espacio muestral.

Traducido a instrucciones sería:

```
300 REM SUBRUTINA PARA EL CACULO DE PORCENTAJES
310 FOR J=1 TO N
320 IF D(J)>(ME-DE) AND D(J)<(ME+DE) THEN LET CN=C
N+1
330 NEXT J
340 PRINT "EL PORCENTAJE ES:";CN*100/N
350 RETURN
```

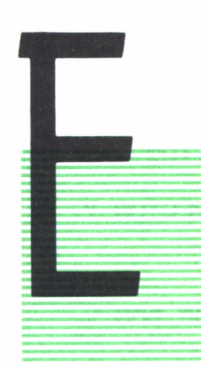
Después de lo visto en estos dos capítulos tenemos todos los elementos necesarios para un análisis completo estadístico.

Supongamos que tenemos los siguientes datos:

```
*****SOLUCION  A  UNOS  DATOS  CLASIFICADOS POR  INTERVALOS******
LOS DATOS SON:
144-148------------------2
148-152------------------15
156-160------------------32
160-164------------------81
164-168------------------90
168-172------------------104
172-176------------------98
176-180------------------76
180-184------------------67
184-188------------------52
188-192------------------19
192-196------------------4
TOTAL--------------------700
```

```
APLICANDO LOS DISTINTOS PROGRAMAS:
LA MODA ES:170.8
LA MEDIA ES:170.7
LA MEDIANA ES:170.7
LA DESVIACION TIPICA ES:10
EL PORCENTAJE ES :97
```

ESPACIO VECTORIAL 15



L estudio de los vectores y su posterior aplicación en la Física hace necesario que se trate en este libro.

Un vector es un segmento orientado, de tal forma que es necesario conocer la dirección, módulo y sentido del mismo.

Lo primero es saber *operar con vectores;* en el siguiente programa se aprenderá a realizar estos procesos:

1.º Cálculo del módulo.
2.º Suma de vectores.
3.º Producto de un vector por un escalar.
4.º Producto escalar.

Para este último punto se calculará, previamente, el valor del ángulo formado por dos vectores. La subrutina correspondiente es:

```
 10 INPUT "COORDENADAS DE LOS DOS VECTORES:",X0,Y0,X1,Y1
500 P1=X0 X1 Y0 Y1
510 P2=SQR(X0^2 X1^2) SQR (Y0^2 Y1^2)
520 COSENO=P1/P2
530 RETURN
```

El programa completo es el siguiente:

```
10 REM OPERACIONES CON VECTORES
20 PRINT "ELIJA OPCION:"
30 PRINT "1-CALCULO DEL MODULO"
40 PRINT "2-CALCULO DE LA SUMA DE DOS VECTORES"
```

```
50 PRINT "3-PRODUCTO DE UN VECTOR POR UN ESCALAR"
60 PRINT "4-PRODUCTO ESCALAR"
70 INPUT OP:IF OP<1 OR OP>4 THEN GOTO 70
80 IF OP=1 THEN GOTO 100
82 IF OP=2 THEN GOTO 200
84 IF OP=3 THEN GOTO 300
86 IF OP=4 THEN GOTO 400
90 END
100 INPUT "INTRODUZCA COORDENADAS DE LOS EXTREMOS:
";X0,Y0,X1,Y1
110 LET MO=SQR(X1-X0)^2+SQR(Y1-Y0)^2)
120 PRINT "EL MODULO ES";MO
130 RETURN
200 INPUT "INTRODUZCA LAS COORDENADAS DE LOS VECTO
RES:";V1,U1,V2,U2
210 PRINT "EL VECTOR SUMA ES:";V1+V2;",";U1+U2
220 RETURN
300 INPUT "INTRODUZCA LAS COORDENADAS DEL VECTOR Y
 EL ESCALAR";V1,U1,A
310 PRINT "EL RESULTADO ES:";V1*A;U1*A
320 RETURN
400 INPUT "INTRODUCIR LAS COORDENADAS DE LOS DOS V
ECTORES:";V1,U1,V2,U2
410 LET M1=SQR(V1^2+U1^2)
420 LET M2=SQR(V2^2+U2^2)
430 REM LLAMADA A LA SUBRUTINA DEL CALCULO DEL COS
ENO INDICADA ANTES
440 GOSUB 500
450 LET PR=M1*M2*CS
460 PRINT "EL PRODUCTO VECTORIAL ES:";PR
470 RETURN
```

Como se ve, el programa principal (10-90) se compone de un menú para la elección de operación. Según ésta serán necesarios unos datos iniciales, introducidos por medio de los input's. Las operaciones en sí no presentan ninguna dificultad, sólo hay que reseñar la llamada a la subrutina del coseno, vista al principio del programa, necesaria para el cálculo del producto escalar.

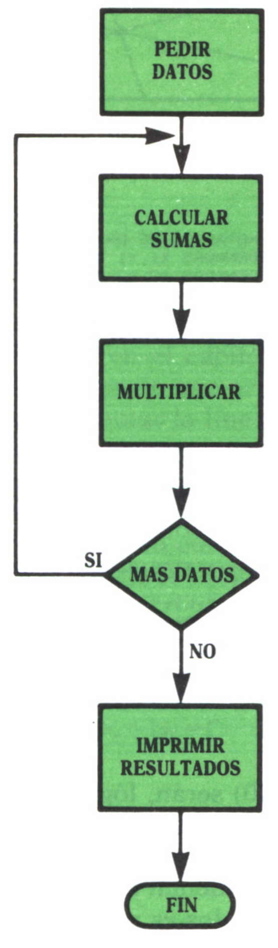
Una vez que hemos aprendido a operar con vectores, dejaremos para el siguiente capítulo los problemas relacionados con bases y cambio de las mismas. Ahora se trata de aprovechar al máximo las coordenadas de un punto.

Dados N puntos, con sus respectivas coordenadas es posible hallar el área del recinto limitado por ellos. Esto es gracias a la fórmula de Gauss. Esta es:

$$AREA = \frac{\Sigma(X(2)+X(1+1)) * (Y(J)-Y(J-1))}{2}$$

La única limitación es la imposibilidad de que se corten los lados, de tal forma que el perímetro se cruzase.

Ver organigrama 18.

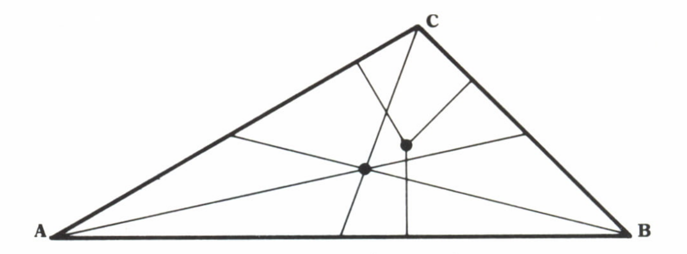


Organigrama 18

Una vez conocida la fórmula para N lados nos centraremos en una figura ya estudiada en este libro: el triángulo.

Conocidos tres puntos, se trata de calcular todos los *puntos notables del triángulo,* tales como: baricentro, incentro, ...

El procedimiento consistirá en ir aplicando las distintas propiedades ya vistas. Todas las fórmulas empleadas son aplicación directa de fórmulas matemáticas.

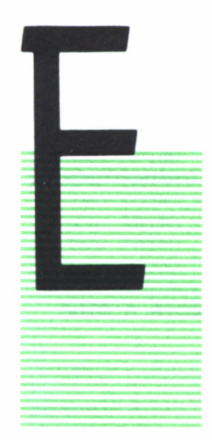


El programa nos resultaría:

```
10 REM CALCULO DE LOS PUNTOS NOTABLES DE UN TRIANGULO
20 INPUT "INTRODUZCA LAS COORDENADAS X1,Y1 DEL PUNTO A: ";X1,Y1
30 INPUT "INTRODUZCA LAS COORDENADAS X2,Y2 DEL PUNTO B: ";X2,Y2
40 INPUT "INTRODUZCA LAS COORDENADAS X3,Y3 DEL PUNTO C: ";X3,Y3
50 LET L1=SQR((X3-X2)^2+(Y3-Y2)^2)
60 LET L2=SQR((X3-X1)^2+(Y3-Y1)^2)
70 LET L3=SQR((X2-X1)^2+(Y2-Y1)^2)
80 LET A1=ATN(L2/L3)
90 LET A2=ATN(L3/L1)
100 LET A3=ATN(L1/L2)
110 LET XB=(X1+X2+X3)/3
120 LET YB=(Y1+Y2+Y3)/3
130 LET XI=(L1*X1+L2*X2+L3*X3)/(L1+L2+L3)
140 LET YI=(L1*Y1+L2*Y2+L3*Y3)/(L1+L2+L3)
150 LET XO=(X1*TAN(A1)+X2*TAN(A2)+X3*TAN(A3))/(TAN(A1)+TAN(A2)+TAN(A3))
160 LET YO=(Y1*TAN(A1)+Y2*TAN(A2)+Y3*TAN(A3))/(TAN(A1)+TAN(A2)+TAN(A3))
170 LET XC=((X2+X3)*TAN(A1)+(X3+X1)*TAN(A2)+(X1+X2)*TAN(A3))/(2*(TAN(A1)+TAN(A2)
+TAN(A3)))
180 LET YC=((Y2+Y3)*TAN(A1)+(Y3+Y1)*TAN(A2)+(Y1+Y2)*TAN(A3))/(2*(TAN(A1)+TAN(A2)
+TAN(A3)))
190 PRINT "LOS LADOS SON: ";L1;" ";L2;" ";L3
200 PRINT "LOS ANGULOS SON: ";A1;" ";A2;" ";A3
210 PRINT "LAS COORDENADAS DEL BAICENTRO SON: ";XB;" ";YB
220 PRINT "LAS COORDENADAS DEL INCENTRO SON:  ";XI;" ";YI
230 PRINT "LAS COORDENADAS DEL ORTOCENTRO SON: ";XO;" ";YO
240 PRINT "LAS COORDENADAS DEL CIRCUNCENTRO SON: ";XC;" ";YC
999 END
```

Los datos iniciales (20-40) serán, lógicamente, las coordenadas de los tres puntos. Los resultados se obtienen aplicando las fórmulas geométricas correspondientes. Partiendo del cálculo de la longitud de los lados (distancia entre dos puntos) se obtienen el baricentro y el incentro (100-130). Gracias al cálculo de los ángulos es posible saber el circuncentro y ortocentro (200-260).

LA RECTA 16



N el capítulo 8 tratamos las ecuaciones de primer grado, posteriormente (en el capítulo anterior) vimos las coordenadas en el espacio vectorial. En éste, uniremos ambos conceptos para tratar la línea recta.

Intentaremos lograr que nuestro ordenador maneje conceptos como paralelismo, perpendicularidad, etc. Como habitualmente hacemos, comenzaremos con lo más sencillo; tengamos en cuenta que nuestro micro no parte con los conocimientos básicos necesarios.

Hay cinco formas comunes de representar a la línea:

1.° Ecuación general:

$$A\times + by = C$$

2.° Forma explícita:

$$Y - Y1 = M(X - X_1)$$

3.° Paramétricas:

$$X = X_1 + T * V_1$$
$$Y = Y_1 + T * V_2$$

4.° Continua:

$$\frac{X - X_1}{V_1} = \frac{Y - Y_1}{V_2}$$

5.° Función segmentos que determina sobre los ejes:

$$\frac{X}{F} + \frac{Y}{G} = 1$$

Lo primero es conseguir saber traspasar la representación de la recta de una a otra. Para ello, según los datos conocidos, pueden darse los siguientes casos:

1.º Conocidos A, B, C.
2.º Conocidos dos puntos (X1, Y1) y (X2, Y2) por los que pasa.
3.º Conocidos un punto (X1, Y1) y el vector dirección (V1, V2).
4.º Conocidos un punto (X1, Y1) y la pendiente M.
5.º Conocidos los segmentos sobre los ejes.

Estas son las posibles opciones para *determinar una recta*.

Confeccionaremos un menú para que, según la opción tecleada, nuestro ordenador reclame los datos necesarios.

En todos los casos hay que conseguir los mismos parámetros A, B, C, V1, V2, M, F, G, X1 y X2; según tengamos unos u otros aplicaremos las relaciones matemáticas oportunas. He aquí el programa; las anotaciones de la derecha te servirán para seguir el desarrollo matemático:

```
10 REM PROGRAMA PARA EL TRASPASO DE UNA FORMULA DE LA RECTA A OTRA
20 PRINT "ELIJA OPCION:"
30 PRINT "1-CONOCIDO A,B,C"
40 PRINT "2-CONOCIDO DOS PUNTOS"
50 PRINT "3-CONOCIDO UN PUNTO Y EL VECTOR DIRECCION"
60 PRINT "4-CONOCIDO UN PUNTO Y LA PENDIENTE"
70 PRINT "5-CONOCIDO LOS SEGMENTOS SOBRE LOS EJES"
80 INPUT OP:IF OP<1 OR OP>6 THEN GOTO 80
90 IF OP=1 THEN GOSUB 200
91 IF OP=2 THEN GOSUB 300
92 IF OP=3 THEN GOSUB 400
93 IF OP=4 THEN GOSUB 500
95 IF OP=5 THEN GOSUB 600
100 GOSUB 700
110 END
200 INPUT "A,B,C: ";A,B,C
210 LET M=-B/A
220 LET X1=1:LET Y1=(A*X1+C)/-B
230 LET X2=2:LET Y2=(A*X2+C)/-B
240 LET V1=X2-X1:LET V2=Y2-Y1
250 LET F=C/A:LET G=C/B
260 RETURN
300 INPUT "INTRODUZCA X1,Y1,X2,Y1 ";X1,Y1,X2,Y2
310 LET V1=X2-X1:LET V2=Y2-Y1
320 LET M=-(Y2-Y1)/(X2-X1)
330 LET A=Y2*(X2-X1)-X1*(Y2-Y1)
350 RETURN
400 INPUT "INTRODUZCA X1,Y1,V1,V2: ";X1,Y1,V1,V2
410 LET M=-V2/V1
420 LET A=V2:LET B=V1
430 LET C=Y1*V1-X1*V2
440 LET F=C/A:LET G=C/B
450 RETURN
500 INPUT "INTRODUZCA X1,Y1,M ";X1,Y1,M
510 LET X2=2:LET Y2=2
```

```
515 LET V2=-V1*M
520 LET A=V2:LET B=-V1
530 LET C=Y1*V1-X1*V2
540 LET F=C/A:LET G=C/B
550 RETURN
600 INPUT "INTRODUZCA LOS DOS SEGMENTOS F,G ";F,G
610 LET C=F:LET A=1:LET B=C/G
620 LET M=-B/A
630 LET X1=1:LET Y1=(A*X1+C)/-B
640 LET X2=2:LET Y2=(A*X2+C)/-B
650 LET V1=X2-X1:LET V2=Y2-Y1
660 RETURN
700 REM IMPRESION
710 PRINT "EN FORMA GENERAL ES:"
720 PRINT A;"X+";B;"Y+";C;"=0"
730 PRINT "EN FORMA EXPLICITA ES:"
740 PRINT "Y-";Y1;"=";M;"(X-";X1;")"
750 PRINT "EN FORMA CONTINUA ES:"
760 PRINT "X-";X1;"/";V1;"= Y-";Y1;"/";V2
770 PRINT "EN FORMA PARAMETRICA ES:"
780 PRINT "X=";X1;"+ T*";V1
790 PRINT "Y=";Y1;"+ T*";V2
800 PRINT "X/";F;"+Y/";G;"=1"
810 PRINT "X/";F;"+Y/";G;"=1"
820 RETURN
```

Por medio del menú principal se bifurca a la opción deseada. En ésta se pedirá los datos necesarios y a partir de éstos se calculan el resto de los datos. En todas las direcciones dadas por el GOSUB se sigue este proceso; las fórmulas se obtienen despejando el dato, hallar de una u otra forma de representación, siempre teniendo en cuenta los datos iniciales.

Obtenida la recta, juguemos con ella; o, mejor dicho, consigamos que nuestro ordenador lo haga. Comencemos con la relación de *paralelismo* y *perpendicularidad* de dos rectas. Para conocer estas relaciones es esencial saber el ángulo que forman. El ángulo formado por una recta y el eje OX viene dado por la pendiente M($M=\tan g\alpha$). Si deseo una recta paralela, ésta deberá tener la misma pendiente. No habría ningún problema en hallarla. Con la perpendicular habría algunas dificultades, ya que habría que aumentar (o disminuir) en M/2 radiones.

Matemáticamente sabemos la condición de que una línea sea perpendicular a otra, es:

$$M1 = \frac{1}{M2}$$

Pongámoselo difícil a nuestro ordenador y programémosle para conseguir la pendiente progresivamente sin aplicar directamente la fórmula.

Supongamos conocidos A, B y C inicialmente (si éstos no fueron los datos, aplicando el programa anterior, los calcularíamos) y nos pidieran calcular la paralela y la perpendicular que pase por el punto (Xo, Yo). Con

la paralela no habría problema y quedará solucionado entre las líneas 70 y 120. Pero la perpendicular la vamos a hallar de diferente manera. Iremos girando la recta paralela obtenida, de 10 en 10 grados, hasta conseguir 90°. En este momento, comprobaremos dos cosas:

1.° Que la recta obtenida tiene su pendiente igual

$$\frac{-1}{M2}$$

2.° La distancia entre el punto (Yo, Yo) y la recta es mínima.

Para este segundo punto seguiremos los siguientes pasos:

a) Calculemos el punto de corte (X1, Y1) entre la recta dato y las diferentes rectas que voy hallando.
b) Calculo la distancia (X1, Y1) y (Xo, Yo).

Como comprobación adicional si en vez de parar el giro en 90° se continúa hasta 100°, observará cómo empieza a aumentar la distancia.

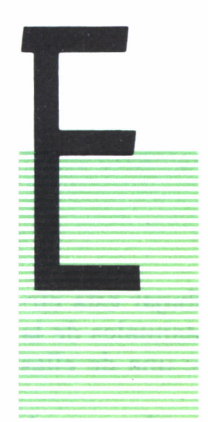
```
10 REM CALCULO DE LA PERPENDICULAR DE UN PUNTO
20 INPUT "INTRODUZCA RECTA DESEADA (A,B,C):";A,B,C
30 INPUT "INTRODUZCA EL PUNTO XO,YO:";XO,YO
40 REM LA FORMULA DE LA RECTA PARALELA VARIA SOLO
EN C
50 LET CP=-(A*XO+B*XO)
60 LET MP=-B/A
70 PRINT "PENDIENTE",,"DISTANCIA"
80 PRINT "=========",,"========="
90 FOR J=1 TO 10
100 LET M1=-(TAN(.1745327777#)+MP)/(TAN(.174532777
#)*MP)
110 LET XC=(M1*B*XO-YO*B-C)/(A+M1)
120 LET YC=-(A*XC+C)/B
130 LET D=SQR((XC-XO)^2+(YC-YO)^2)
140 PRINT M1,D
150 MP=M1
160 IF J=9 THEN MS=M1
170 NEXT J
180 PRINT "LA ECUACION ES :"
190 PRINT "Y=";YO;"=";M1;"(X-";XO;")"
999 END
```

El cálculo de la paralela (líneas 50, 60) no representa ninguna dificultad al aplicar directamente las fórmulas de paralelismo. Con el FOR (línea 90-120) se produce el proceso de girar la línea 100 grados, de 10 en 10.

calculando la distancia (98) a la vez que la pendiente (95). Para saber la solución se utiliza la instrucción 110.

Si se atreve, la versión gráfica de este programa puede serle interesante. Sólo tiene que ir tomando como referencia el punto de corte y lanzar líneas (por ejemplo, DRAW en el Amstrad) desde (Yo, Yo) a los distintos puntos de corte.

CURVAS CONICAS 17

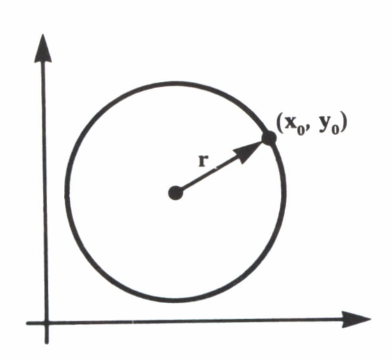


EL estudio de las cónicas con sus relaciones es el tema a tratar en este capítulo. Empezaremos con la más conocida de todas ellas: *la circunferencia*.

El lugar geométrico de los puntos del plano que equidistan de un punto fijo llamado centro es la circunferencia.

¿Qué es un lugar geométrico?

Es el conjunto de puntos que poseen una determinada propiedad (y además son los únicos puntos que la poseen). Un dato a tener en cuenta es el radio. Definido como la distancia del centro a cualquier punto de la circunferencia.



El radio vendrá dado por la expresión: $r=\sqrt{^2+Y^2}$. Elevando al cuadrado se obtendrá la fórmula general de la circunferencia.

$$R^2=X^2+Y^2$$

El siguiente programa calculará los valores de los puntos de la circunferencia. Como todos no se pueden hallar (son infinitos), impondremos un intervalo 5/100.

```
10 REM CALCULO DE LA TABLA DE VALORES DE UNA CIRCU
NFERENCIA
20 INPUT "INTRODUZCA RADIO: ";R
30 PRINT "X"," Y"
40 PRINT "-"," -"
50 LET DE=2*3.14159/100
60 FOR J=1 TO 20
70 LET X=R*COS(DE*(J-1))
80 LET Y=R*SIN(DE*(J-1))
90 PRINT X,Y
100 NEXT J
```

Ya hemos conseguido hallar una tabla de valores. El dibujar esta circunferencia lo dejo al lector. Tenga en cuenta que en los Manuales de la mayoría de los micros vienen unas subrutinas de ejemplos; una de ellas suele ser el dibujo de la circunferencia.

La circunferencia puede venir dada por la fórmula llamada general:

$$AX^2+BY^2+CXY+DX+EY+F=0$$

Las condiciones que deben cumplir estos coeficientes son:

$$A=1.K \qquad D=-2ak$$
$$B=1.K \qquad E=-2bk$$
$$C=0.K \qquad F=(a^2+b^2-r^2)\ K$$

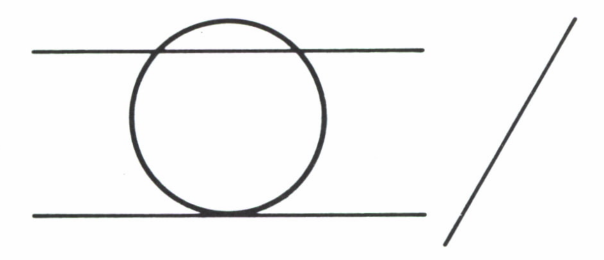
Si despejamos a, b y r nos queda:

$$a=-\frac{D}{2A} \quad b=-\frac{E}{2A} \quad r=\frac{1}{2A} \quad \sqrt{D^2+E^2-4AK}$$

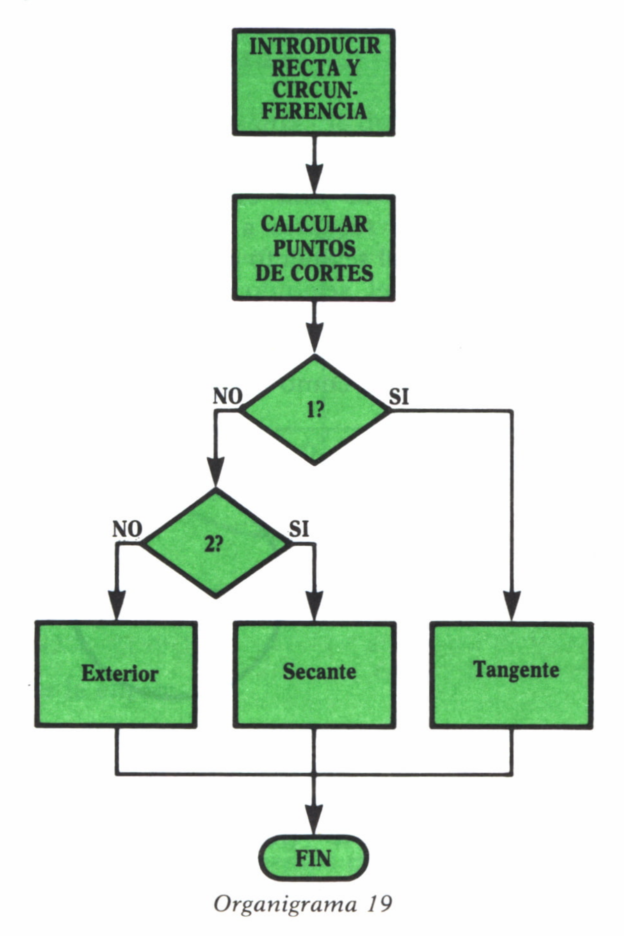
Es decir, será una circunferencia de centro

$$C\left(\frac{D}{-2A},\frac{E}{-2A}\right)$$

Dada una recta puede ser interesante saber su *posición respecto a una circunferencia*. Fíjese en la figura y donde pueden verse los tres posibles casos:



Ver organigrama 19.



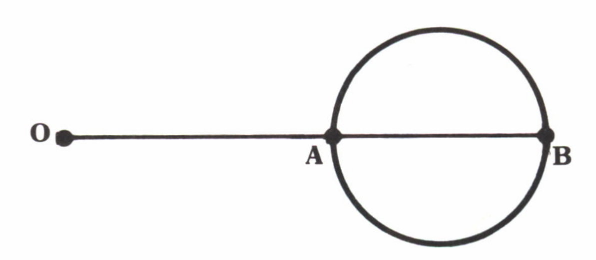
Organigrama 19

La lista de instrucciones no revistirá ninguna dificultad. Simplemente habrá que resolver el sistema dado por las ecuaciones de las dos figuras geométricas: recta y circunferencia. Nos quedará una ecuación de segundo grado, y analizando su discriminante se obtendrá la posición de la recta.

```
10 REM CALCULO DE LA POSICION DE UNA RECTA EN UNA
CIRCUNFERENCIA
20 INPUT "INTRODUZCA EL RADIO : ";R
30 INPUT "INTRODUZCA COEFICIENTES DE LA RECTA (A,B    ):"
,C) :";A,B,C
40 LET DE=2*3.14159/100
50 REM X^2+(-C-AX)^2=R^2. DESARROLLANDO QUEDA: (A^
2+1)X+2CX+(C^2-R^2)                                  1)X
60 LET B=2*C:LET A=A^2+1:LET C=C^2-R^2
70 LET DI=B^2-4*A*C
90 IF DI<0 THEN PRINT "LA RECTA ES EXTERIOR."
100 IF DI=0 THEN PRINT"LA RECTA ES TANGENTE"
110 IF DI>0 THEN PRINT "LA RECTA ES SECANTE"
999 END
```

Un concepto importante en el estudio de las circunferencias es la *potencia de un punto respecto de una circunferencia*. Dado un punto P(Xo, Yo), si de él se trazan secantes a la circunferencia se cumple: PA.PB=constante, siendo PA el segmento que une P y el primer punto de corte, y PB lo mismo con el segundo punto de intersección de la circunferencia.

A este producto se llama potencia de un respecto a una circunferencia. Si logramos trazar un secante tal como la figura:



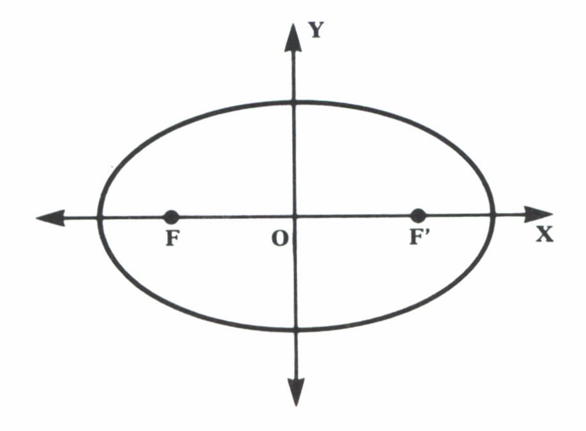
Se ve que $P=d^2-r^2$, ya que $PA=d-r$ y $PB=d+r$ siendo d la distancia del punto a la recta, sabiendo que $d^2=(Xo-a)^2+(Yo-b)^2$, la potencia será:

$$P_{(o)}=(Xo-a)^2+(Yo-b)^2-r^2$$

Dadas dos circunferencias se llama *eje radical* a la recta compuesta por los puntos que tienen la misma potencia respecto a las dos circunferencias. Dado los centros y radios de dos circunferencias calcularemos su eje radical al siguiente programa:

```
10 REM CALCULO DEL EJE RADICAL DE DOS CIRCUNFERENC
IAS
20 INPUT "INTRODUZCA CIRCUNFERENCIA 1 EN FORMA GEN
ERAL : ";D1,E1,F1
30 INPUT "INTRODUZCA CIRCUNFERENCIA 2 EN FORMA GEN
ERAL : ";D2,E2,F2
40 PRINT "EL EJE RADICAL DE LAS DOS CIRCUNFERENCIA
S ES:"
50 PRINT "COEFICIENTE DE LA X: ";D1-D2
60 PRINT "COEFICIENTE DE LA Y: ";E1-E2
70 PRINT "TERMINO INDEPENDIENTE: ";F1-F2
999 END
```

Otra cónica conocida es la *elipse*. Se define como el lugar geométrico de los puntos, tales que la suma de las distancias a dos puntos fijos llamados focos en constante.



Elementos de la elipse:

F y F: Focos de la elipse.
Eje focal: Eje que pasa por los focos.
Eje secundario: Eje mediatriz del segmento $\overline{FF}$.
Vértices: Puntos donde cortan los ejes a la elipse.
Radios vectores: Son los segmentos que unen los focos con un punto cualquiera de la elipse.

La fórmula de la elipse vendrá dada por:

$$\frac{(X-X)^2}{a^2} \times \frac{(X-Yo)^2}{b^2} = 1$$

donde Xo y Yo son las coordenadas del centro.

Veamos cómo podemos imprimir una elipse. Para ello reduzcamos la fórmula a:

$$\frac{X^2}{a^2} + \frac{Y^2}{b^2} = 1$$

Hay que tener en cuenta que en la impresora, al menos las usuales, no pueden subir y bajar. El gráfico habrá que hacerlo de tal forma que vayan saliendo las líneas horizontalmente. Es decir, la elipse quedará como un balón de rugby. Al aplicar la fórmula, ésta nos da dos posibles valores de y:

```
10 REM TABLA DE VALORES DE UNA ELIPSE
20 INPUT "INTRODUZCA VALOR DE LOS SEMIEJES A,B: ";
A,B
30 PRINT " X"," Y1"," Y2"
40 PRINT " -"," --"," --"
50 LET I=B/20
60 FOR J=1 TO 21
65 LET X=I*(J-1)-A
70 LET Y1=(C-(B^2)*(X^2))/A^2
80 LET Y2=(C-(B^2)*(X^2))/A^2
90 PRINT X,Y1,Y2
100 NEXT J
999 END
```

Partiendo desde X=A hasta X=0 tendremos media elipse dibujada. Si lo hiciéramos desde X=− a hasta X=a obtendríamos la elipse entera. Se va a hacer sólo media elipse, para después repetir el FOR de 0 a A, consiguiendo la otra media.

El programa no tiene dificultad, bastará controlar los blancos antes y después de imprimir los asteriscos.

¿Dónde estarán los ejes?

Debido al problema de la impresora, los ejes quedarán invertidos.

Otro punto a tener en cuenta es la limitación de nuestra gráfica. Podemos disponer de un ancho de 80 columnas, luego el valor máximo de Y será 39 (80−2/2). Esta cota habrá que tenerla en cuenta a la hora de saber el valor de cada posición de la impresora.

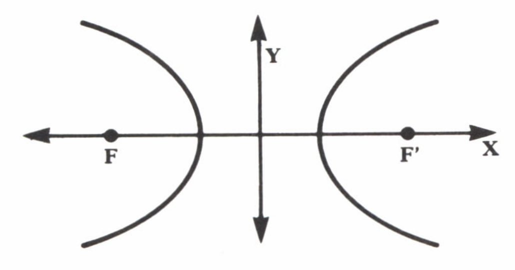
¿Cómo calcularla?

Muy sencillo, el valor máximo de Y es cuando X=0, sustituyendo este valor en la fórmula: Y=

Si como mucho el ancho puede ser B, cada posición representará un valor de: (39/B). Tampoco sirven los decimales, luego habrá que utilizar la función INT. Esto originará pequeños relieves en nuestra gráfica. El programa quedaría:

```
10 REM GRAFICA DE LA ELIPSE
20 INPUT "INTRODUZCA VALOR DE LOS SEMIEJES A,B: ";
A,B
30 LET C=A^2*B^2
40 LET I=B/20
50 FOR J=1 TO 21
60 LET X=I*(J-1)-A
70 LET Y1=(C-(B^2)*(X^2))/A^2
75 LET Y1=SQR(ABS(Y1))
80 GOSUB 200
90 NEXT J
100 END
200 REM SUBRUTINA DE GRAFICA
205 IF Y1=0 THEN PRINT "
*":GOTO 310
210 LET P=39/(B*2)
220 LET G=INT(Y1*P)
225 LET G=ABS(G):LET N=39-G
226 PRINT
230 FOR B=1 TO N
240 PRINT " ";
250 NEXT B
260 PRINT "*";
265 LET N=2*G
270 FOR B=1 TO N
280 PRINT " ";
290 NEXT B
300 PRINT "*";
310 RETURN
```

Ya sabe el ordenador manejar circunferencias y elipses, completemos su formación con la *hipérbola*. Esta viene representada por la figura:



Su definición es: lugar geométrico de todos los puntos del plano tales que la diferencia de las distancias a dos puntos fijos llamados focos es una cantidad constante.

Elementos significativos:

Focos K y F', son los puntos fijos de la definición.

Distancia focal: $\overline{FF.}$

Radios vectores: $\overline{PF}$ y $\overline{PF'}$.

Eje focal: recta que pasa por los focos.

Se cumple que para cualquier punto P:

Su fórmula general:

$$\frac{(X-Xo)^2}{a^2} - \frac{(Y-Yo)^2}{b^2} = 1$$

Siendo (Xo, Yo) los coordenados del centro de la hipérbola.

Al igual que se hizo con las otras dos cónicas se programará al ordenador para obtener la fórmula según unos datos dados.

```
10 REM CALCULO DE LA FORMULA GENERAL DE LA HIPERBO
LA
20 INPUT "INTRODUZCA COORDENADAS DEL CENTRO (X,Y):
 ";X,Y
30 INPUT "INTRODUZCA VALORES DE A Y B: ";A,B
40 PRINT "EL TERMINO DE X^2 ES: ";B^2
50 PRINT "EL TERMINO DE Y^2 ES: ";-(A^2)
60 PRINT "EL TERMINO DE X ES  : ";-(X*B^2)
70 PRINT "EL TERMINO DE Y ES  : ";Y*A^2
80 PRINT "EL TERMINO INDEPENDIENTE ES : ";(B^2*X^2
)-(A^2*B^2)-(A^2*Y^2)
```

Habiendo comprendido la circunferencia y la elipse no entraña dificultad alguna el programa anterior.

Un caso interesante es cuando dan una ecuación y piden decir qué clase de cónica es.

Para ello, se basará el razonamiento en la siguiente propiedad matemática:

Sea $AX^2+BXY+CY^2+DX+EY+F$

Se hace: $A*C*F-2*B*E-A*E^2-C*D^2-F*B^2=DETER$

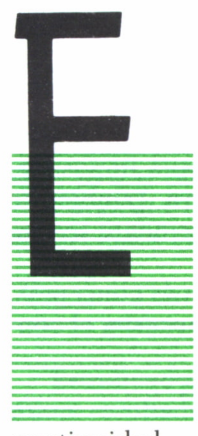
Si DETER >0 es hipérbola
DETER <0 es elipse

El programa deducido será:

```
10 REM SUBRUTINA PARA DISTINGUIR DOS CONICAS
20 PRINT "INTRODUZCA DATOS:"
30 INPUT "EL TERMINO EN X^2 ES :";A
40 INPUT "EL TERMINO EN X*Y ES :";B
50 INPUT "EL TERMINO EN Y^2 ES :";C
60 INPUT "EL TERMINO EN X ES   :";D
70 INPUT "EL TERMINO EN Y ES   :";E
80 INPUT "EL TERMINO INDEPENDIENTE ES: ";F
90 LET DE=A*C*F-2*B*D*E-A*E^2-C*D^2-F*B^2
100 IF DE>0 THEN PRINT "ES UNA PARABOLA"
110 IF DE<0 THEN PRINT "ES UNA ELIPSE"
120 IF DE=0 THEN PRINT "ES UN CIRCULO"
999 END
```

Queda para el lector el intentar imprimir la hipérbola y la circunferencia, el proceso es similar al de la elipse. Después de este capítulo nuestro ordenador domina la geometría... o al menos es menos torpe de lo habitual.

ESTUDIO DE FUNCIONES 18



L estudio de las gráficas de las funciones es cuestión de experiencia y suerte. La primera es imprescindible para aplicar a simple vista los conocimientos teóricos en los puntos adecuados, evitando, de esta manera, pruebas con valores innecesarios.

Suerte, sobre todo al estudiar la *continuidad*. En determinadas funciones la discontinuidad está clara; este es el caso, por ejemplo, de las funciones con denominadores. Hallando las raíces de éstos, ya se tienen los puntos de continuidad.

El ordenador no puede, al menos en principio, intuir qué método a aplicar para estudiar la continuidad. La manera de averiguarlo es ir probando valores; cuando el ordenador, no el programa, dé error de operación habremos encontrado un punto de discontinuidad. Lógicamente al interrumpir el proceso, es necesario saber con qué valor ha ocurrido esto; para ello introducimos una instrucción PRINT antes de operar. Si se diese el caso de hallar uno, podríamos seguir buscando reanudando el proceso con límite inferior, el valor de X siguiente al hallado.

Este método rudimentario de buscar el crack del ordenador tiene en principio dos defectos:

1.º Obliga al operador a estar atento a la ejecución apuntando los puntos de discontinuidad.

2.º Puede ocurrir que aunque elijamos un incremento de X pequeño, el punto buscado puede estar entre dos valores de prueba, de tal forma que nos saltaríamos este punto y no sería detectado.

Esta segunda dificultad puede presentarse aunque no se emplee el ordenador; por tanto, no es toda la culpa suya.

De todas formas, a no ser que se analice previamente la función, es un método válido inicialmente.

```
10 REM ESTUDIO DE CONTINUIDAD
20 DEF FNA(X)=(X^2-1)
30 INPUT "INTRODUCIR LIMITES DEL INTERVALO: ";A,B
40 INPUT "INTRODUCIR NUMERO DE SUBINTERVALOS A REA
LIZAR: ";L
50 LET X=A
60 PRINT X,
70 LET F=FNA(X)
75 PRINT F
80 LET X=X+(B-A)/L
90 IF X<B THEN GOTO 60
999 END
```

Se ve cómo se puede elegir el intervalo a tratar, ya que los estudios de una función se hacen respecto a un intervalo concreto; sería imposible, quitando excepciones, un estudio exhaustivo de toda la curva. También se puede escoger el 0 incremento de X; a medida que el incremento sea inferior, mayor será la veracidad del estudio.

Una vez estudiada la continuidad seguiremos con los demás conceptos. Tanto para máximo, mínimo, puntos de inflexión, etc., se necesita el concepto de derivada.

La *derivada* de una función en un punto viene dada por:

$$f'(Xo) = \lim_{h-Po} \frac{f(Xo)-f(X+h)}{h}$$

Los límites, por el momento, no son asequibles al micro; para resolver este inconveniente se utilizará un método aproximativo. Partiendo de una H determinada, e ir decrementando este valor, se obtendrá una tabla de valores, de tal forma que llegará un valor de H en el que la diferencia de la derivada con la anterior hallada sea prácticamente nula. A partir de aquí se tomará como derivada el último hallado. Se utilizará como cota de la diferencia el valor 0,05, para no multiplicar en exceso los cálculos.

Como siempre, para obtener la derivada de diferentes funciones, bastará modificar las instrucciones donde se aplique la función. Seguidamente se observará la estructura del proceso; la lista de las instrucciones da la subrutina que se empleará para el cálculo de la derivada en un punto.

```
10 REM ESTUDIO DE LA DERÍVADA EN UN PUNTO
20 DEF FNA(X)=(X^2-1)
30 REM PETICION DE NUMERO PARA CALCULAR SU DERIVADA
```

```
40 INPUT "INTRODUCIR NUMERO: ";P
50 LET H=1:LET AN=0
60 LET X0=P
70 LET X1=X0+H
80 LET DE=(FNA(X1)-FNA(X0))/(X1-X0)
90 IF ABS(AN-DE)<.005 THEN GOTO 140
100 PRINT X0,X1,DE
110 LET AN=DE
120 LET H=H/2
130 GOTO 70
140 PRINT "LA DERIVADA ES : ";DE
150 END
```

Naturalmente, antes de probar un Xo habrá que estudiar su continuidad, ya que ésta es condición necesaria para la existencia de la derivada.

Conocido el proceso de la derivada pasaremos a estudiar la función. Empecemos viendo si una función es *creciente* o *decreciente*.

Sea

$$f(X) = X^2 - 1$$

Estudiando el proceso en el intervalo [A, B] iremos comparando los distintos resultados f(X), viendo cómo evolucionan. Se observará los distintos casos en procesos independientes, para no complicar el razonamiento (ver organigrama 20).

La lista de instrucciones quedaría:

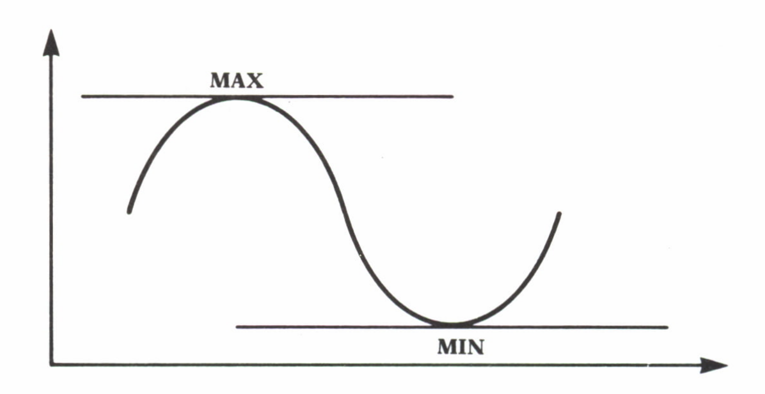
```
10 REM ESTUDIO DE UNA CURVA PARA VER SI ES CRECIENTE O DECRECIENTE
20 DEF FNA(X)=(X^2-1)
30 INPUT "INTRODUCIR LIMITES INTERVALOS :";A,B
40 INPUT "INTRODUCIR INCREMENTO DE X: ";L
50 LET N=INT(B-A)/L
60 DIM A(N)
70 FOR J=1 TO N
80 LET A(J)=FNA(A+L*(J-1))
90 NEXT J
100 REM ESTUDIO SI ES CRECIENTE
110 LET CN=0
120 FOR J=2 TO N
130 IF A(J)>A(J-1) THEN GOTO 160
140 IF A(J)=A(J-1) THEN LET CN=1:GOTO 160
150 GOTO 200
160 NEXT J
170 IF CN=0 THEN PRINT "ES ESTRICTAMENTE CRECIENTE"
180 IF CN=1 THEN PRINT "ES CRECIENTE"
190 GOTO 999
200 LET CN=0
```

```
210 FOR J=2 TO N
220 IF A(J)>A(J-1) THEN GOTO 250
230 IF A(J)=A(J-1) THEN LET CN=1:GOTO 250
240 GOTO 999
250 NEXT J
260 IF CN=0 THEN PRINT "ES ESTRICTAMENTE DECRECIENTE"
270 IF CN=1 THEN PRINT "ES DECRECIENTE"
999 END
```

El uso de la variable de control es para definir si es estrictamente creciente o decreciente.

Ya se ha logrado avanzar otro paso para que el ordenador estudie una función. La dimensión de la matriz será (B−A)/L|, habrá que procurar que L sea divisor de (B−A) para no desperdiciar datos.

Otros datos interesantes son los *máximos* y *mínimos* en un intervalo. Fíjese en las gráficas siguientes:

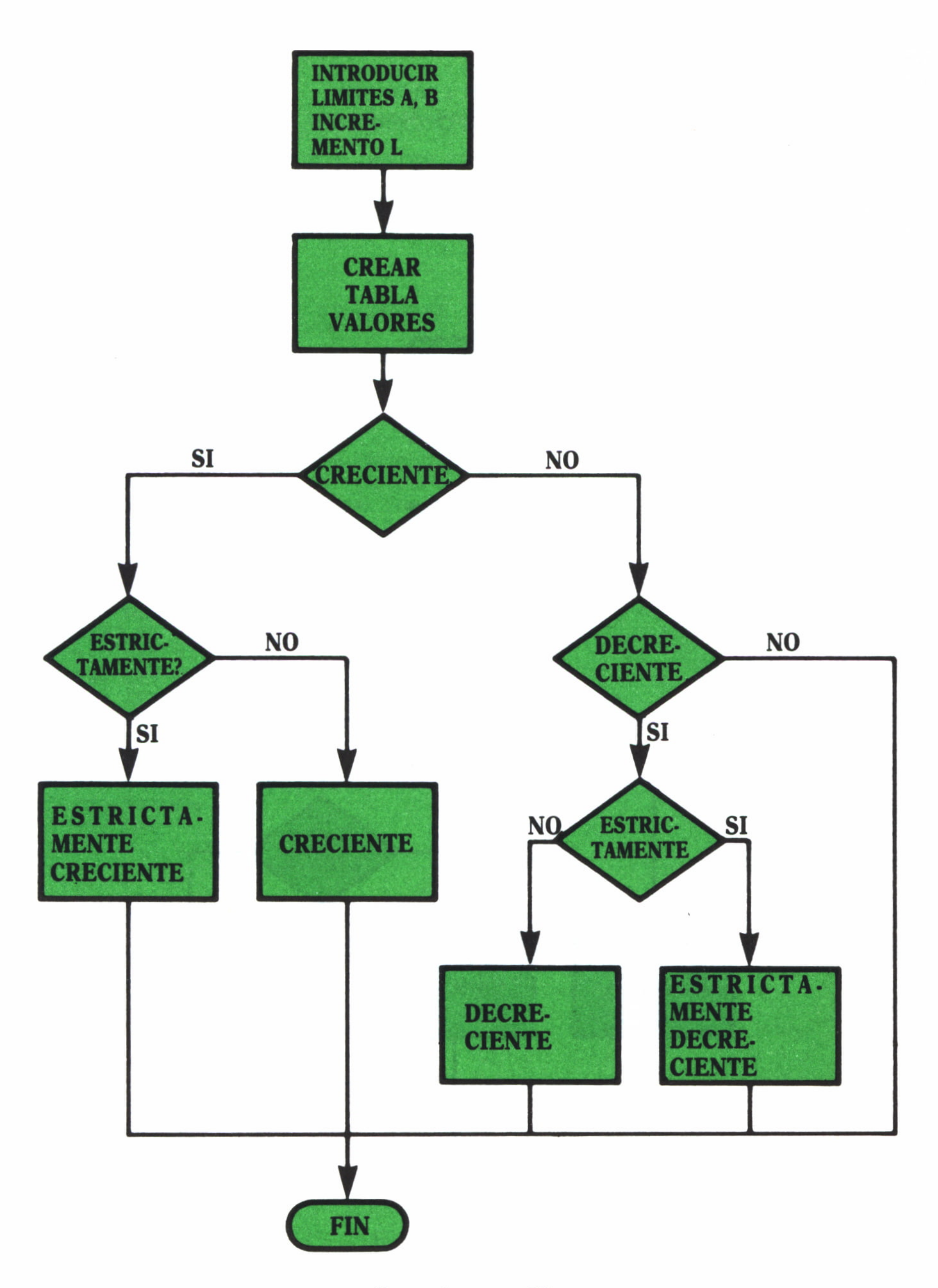


Se observa que en un punto el valor de X es máximo o mínimo si la pendiente en ese punto de la tangente a la curva es nula. Para ello bastará que la derivada de la curva en ese punto sea 0. El sistema a seguir es estudiar la evolución de los valores de la derivada. Si cambia de signo querrá decir que existe un punto donde se anula la derivada.

Ya se planteó en el capítulo 8 (la resolución de una ecuación) cómo se ve el punto de corte reduciendo el intervalo formado por un valor negativo y otro positivo, o viceversa.

Encontrando este punto habrá que averiguar si es máximo o mínimo; para esta labor, se observará la evolución del signo.

Si la pendiente va de negativa a positiva, es un mínimo; en caso inverso es un máximo.

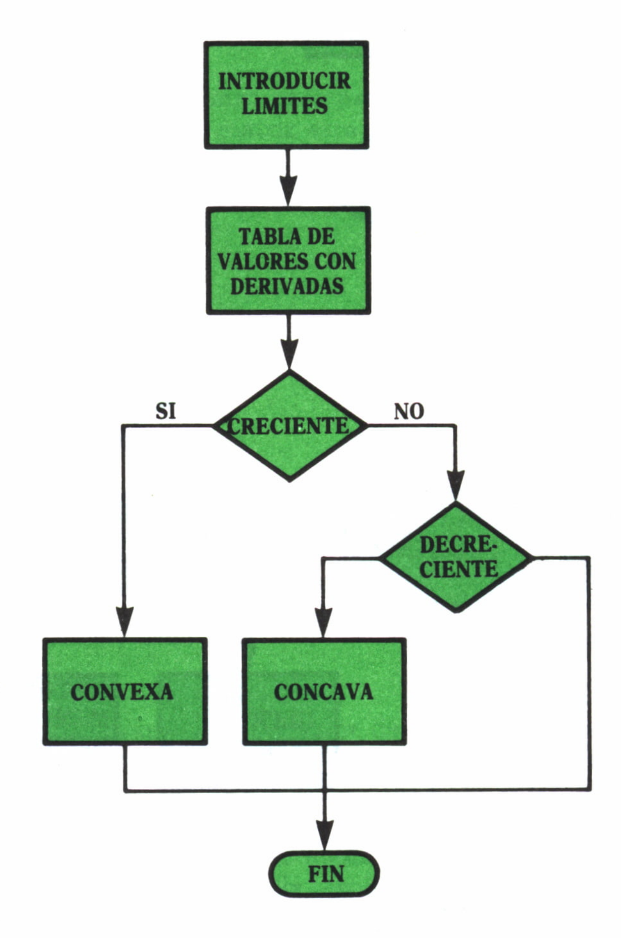


Organigrama 20

Para observar la *convexidad* y *concavidad* de una curva hay que tener en cuenta los siguientes criterios:

— Una función es convexa si f' es creciente.
— Una función es cóncava si f' es decreciente.

Si para ver los máximos y mínimos se estudió el signo de la derivada, para ver la convexidad y concavidad haremos un proceso similar al estudio de ver si una función es creciente o decreciente al principio del tema. Ver organigrama 21.



Organigrama 21

Si no es cóncava ni convexa en el intervalo [A, B], quiere decir que existen unos varios puntos en los que cambia; estos son los puntos de inflexión.

El programa para ver si es cóncava o convexa es:

```
10 REM ESTUDIO DE UNA CURVA PARA VER SI ES CONCAVA O CONVEXA
20 DEF FNA(X)=(X^2-1)
30 INPUT "INTRODUCIR LIMITES INTERVALOS : ";L
50 LET N=INT(B-A)/L
60 DIM A(N)
70 FOR J=1 TO N
75 GOSUB 500
80 LET A(J)=VA
90 NEXT J
100 REM ESTUDIO SI ES CONVEXA
110 FOR J=2 TO N
120 IF A(J)=A(J-1) THEN GOTO 140
130 GOTO 200
140 NEXT J
150 PRINT "ES CONVEXA"
160 GOTO 999
200 REM ESTUDIO SI ES CONCAVA
210 FOR J=2 TO N
220 IF A(J)=A(J-1) THEN GOTO 240
230 GOTO 999
240 NEXT J
250 PRINT "ES CONCAVA"
260 GOTO 999
500 REM CALCULO DE LA DERIVADA
510 LET X0=A+J*L:LET H=.5
515 LET X1=X0+H
520 LET DE=(FNA(X1)-FNA(X0))/(X1-X0)
530 IF ABS(DE-AN)<.005 THEN GOTO 570
540 LET AN=DE
550 LET H=H/2
560 GOTO 515
570 LET VA=INT(DE*1000)/1000
580 RETURN
999 END
```

El lector puede intentar el cálculo de los puntos de inflexión. Para ello, deberá encontrar el punto donde cambia de creciente a decreciente, o viceversa. ¡Inténtelo!

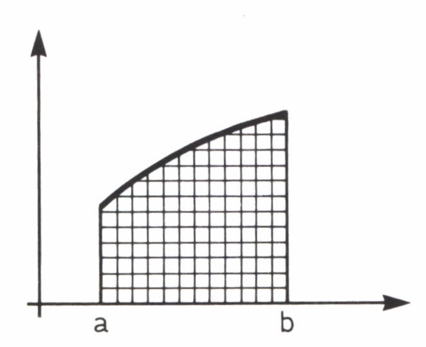
AREA DEL RECINTO LIMITADO POR UNA CURVA. INTEGRALES 19



SI nos pidieran calcular el área limitada por una curva, a más de uno nos pondrían en un terrible aprieto. Rápidamente intentaríamos hacer memoria de todas las reglas de integración, pero en la mayoría de los casos conseguiríamos un terrible dolor de cabeza.

Entonces la forma más fácil es, como anteriormente, pasárselo al ordenador.

En principio, el ordenador «no sabe» integrar. Enseñémosle a hacerlo.



Si dividimos el intervalo en rectángulos iguales, la suma de estos rectángulos se aproximaría, más o menos, al área de la curva. Los rectángulos exteriores darían un valor mayor que el de la curva y los interiores se quedarían «cortos». Si disminuimos el ancho de estos intervalos la diferencia se haría más pequeña. Repitamos este proceso varias veces consiguiendo que el rectángulo sea prácticamente una línea, reduciendo la base de ellos; habremos conseguido una aproximación casi exacta al área de la curva.

En este momento, los rectángulos exteriores serían prácticamente iguales a los interiores.

Según este razonamiento, para «construir» nuestro programa debemos tener en cuenta:

a) Cuáles son los límites de integración.

b) Hemos de controlar razonadamente la disminución de las bases. Para ello dividiremos el intervalo completo en dos intervalos iguales, repitiendo progresivamente el proceso con intervalos que sean la mitad de los anteriores. De esta manera el primer cálculo sería con dos rectángulos, después 4, 6, 8 ..., así hasta conseguir que la longitud de los intervalos sea prácticamente nula.

c) La altura de estos rectángulos vendrá dada por el valor de la función en el extremo superior del intervalo. Nos estamos basando en los rectángulos exteriores para la resolución del problema; el razonamiento sería semejante con los interiores. Por tanto, nuestro programa valdrá para cualquier función, sólo bastará rectificar la instrucción donde se calcule la función.

Para comprender mejor todo el proceso vamos a escoger una función sencilla, por ejemplo: $f(X)=X^2$ en el intervalo [a, b].

Ahora fijémonos en el organigrama y la lista de instrucciones siguiente, inicialmente sólo lo que aparece en el interior de los recuadros sombreados.

Según nuestro razonamiento, lo primero sería introducir los datos iniciales:

INPUT "Introducir límites integración: "; a, B.

Entramos ahora en el proceso repetitivo del cálculo del área. Hemos dicho que calcularíamos el área del recinto como suma de las áreas de los rectángulos circunscritos. Ahora bien, este proceso habrá que repetirlo sucesivamente hasta conseguir una aproximación aceptable.

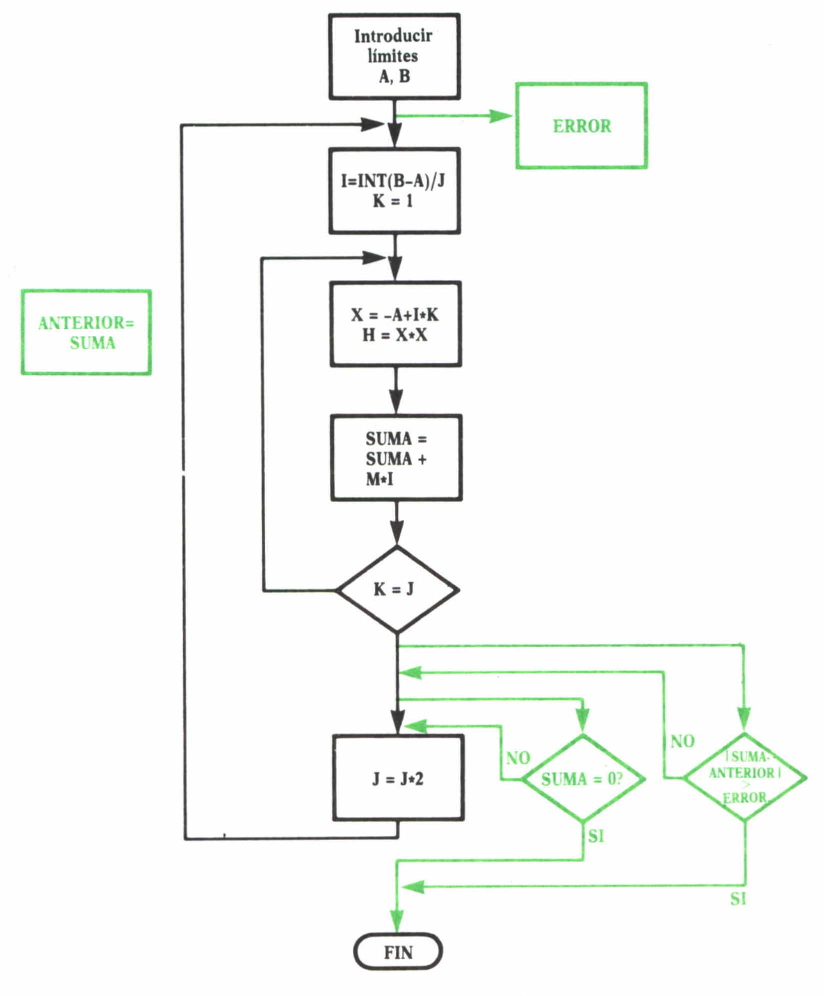
Para obtener la suma en cada caso es necesario conseguir unos parámetros básicos:

```
116  SUMA=0
120  J=J*2
```

En la 116 inicializamos el valor de suma en la 120 y obtenemos el valor del intervalo, es decir, la longitud de cada una de las bases de los rectángulos. Observamos que J se duplica, luego I se reducirá a la mitad cada vez que iniciemos el proceso.

Con la instrucción **FOR**, lo que hacemos es calcular el área de cada rectángulo e ir acumulando (**SUMA**) hasta obtener el valor del área total. Si nos fijamos, la altura (H) viene dada por el valor de la función en ese punto. Es aquí, en la 70, donde tendríamos que modificar para obtener la in-

tegral de cualquier otra función; la base, lógicamente, nos la da la longitud del intervalo (I) (ver organigrama 22).



Organigrama 22

```
10 REM CALCULO DEL AREA DEFINIDA POR UNA CURVA
20 INPUT "INTRODUCIR LIMITES DE LA INTEGRACION: ";A,B
25 INPUT "INTRODUCIR COTA DE ERROR: ";CO
30 LET J=2:LET SU=0
35 PRINT " J"," I"," SUMA"
```

```
40 LET I=INT(B-A)/J
50 FOR K=1 TO J
60 LET X=A+I*K
70 LET H=X*X
80 LET SU=SU+(H*I)
90 NEXT K
100 IF ABS(SU-AN)<CO THEN GOTO 140
110 LET AN=SU
115 PRINT J,I,SU
116 LET SU=0
120 LET J=J*2
130 GOTO 40
140 PRINT "EL VALOR DEL AREA ES: ";SU
```

Una vez concluida la suma, imprimimos los resultados y volvemos a la 115 a repetir el proceso.

Se reiterará el proceso indefinidamente, ya que no hemos puesto ningún «control de fin».

¿Cómo terminar? Podríamos tomar varias opciones, pero vamos a tratar dos.

La primera de ellas (color verde) se basa en el hecho de que al aumentar J, disminuye I, llegando el momento que I fuese tan pequeño que la máquina lo considerase 0. En este momento, todas las áreas siguientes valdrían 0, luego el valor conseguido antes de este momento sería el buscado.

Por ello, no tenemos más que preguntar si la SUMA=0. Terminando el programa en el caso de cumplirse. Podríamos haber hecho la pregunta después de la XXX consultando el valor de I.

La otra (color rojo) es poder nosotros elegir el final. Introducimos una cota de error y cuando la diferencia entre la suma y la suma anterior es menor que esta cota se termina el proceso. El porqué de esta opción se debe a que llega un momento que los rectángulos son tan estrechos que la diferencia área total es tan pequeña que no merece la pena seguir.

Analicemos los resultados del programa:

```
LOS RESULTADOS DEL PROGRAMA SON:
INTRODUCIR LIMITES DE LA INTEGRACION: 0,1
INTRODUZCA COTA DE ERROR :0.005
J               I               SUMA
-               -               ----
2               0.5             0.625
4               0.25            0.46875
8               0.125           0.3984375
```

```
16                 0.0625            0.365234375
32                 0.03125           0.349121094
64                 0.015625          0.341186523
EL VALOR DEL AREA ES : 0.33724756
```

Calculamos ahora la integral por el método tradicional:

$$\int_{0}^{1} X^2 = \left[\frac{X^3}{3}\right]_{0}^{1} = \frac{1}{3} = 0{,}33$$

Hemos comprobado que nuestro ordenador es capaz de hacer una integral definida. Ahora bien, ¡cuidado!, ya que antes de hacerla habrá que analizar esta integral.

Debemos actuar como si la hiciéramos nosotros, es decir, ver los puntos de corte, continuidad... Nuestro ordenador es «listo», pero nunca podrá hacer más de lo que le «enseñemos».

Se repasan los principales conceptos de la ciencia citada, desde un punto de vista eminentemente práctico y para su aplicación al ordenador personal. Se basa el texto en la presentación de pequeños programas (que usted podrá introducir en su ordenador personal); estos programas se analizan y se indican posibles modificaciones a introducir para variar los resultados que se deben obtener. Libro útil tanto para que el estudiante repase estos conceptos que está aprendiendo como para que el adulto recuerde algunos conocimientos olvidados que le interesan hoy.